Energy efficiency and energy balance are two important issues for wireless sensor networks. In previous clustering routing algorithms, multihop transmission, sleep scheduling, and unequal clustering are always used to improve energy efficiency and energy balance. In these algorithms, only the cluster heads share the burden of data forwarding in each round. In this paper, we propose a flow-partitioned unequal clustering routing (FPUC) algorithm to achieve better energy efficiency and energy balance. FPUC consists of two phases: clustering and routing. In the clustering phase, the competition radius is computed according to the node density and the distance from sensor nodes to the sink. The sensor nodes that have more residual energy and larger overlapping degree have higher probability to be selected as cluster heads. In the routing phase, each cluster head first finds the gateway nodes and then distributes the data flow to each of its gateway nodes depending on residual energy. After that, each gateway node forwards the data to the next hop with minimum cost. Two metrics called network lifetime and coverage lifetime are used to compare the performance of FPUC with that of the existing ones. Simulation results show that FPUC can achieve longer network lifetime and coverage lifetime than previous algorithms.
the CH will also have effects on its energy consumption, especially under the nonuniform distribution. As Figure 1 shows, areas $a$ and $b$ have approximately equal distance to the sink, but node density of area $a$ is obviously larger than that of area $b$ which will lead to those CHs in area $a$ having heavier burden than CHs in area $b$, and thus CHs in area $a$ will consume more energy and then the energy hole may occur.

In [12–15], researchers have proposed some algorithms which aim to conserve energy by turning off as many as possible sensor nodes while achieving the application’s coverage goal at the same time. Tao et al. first propose the flow-balanced routing protocol (FBR) [16], which is very different from traditional routing algorithms. In FBR, a graph topology is constructed instead of tree structure and each CH may have multiple parents. When CH relays data, it computes the amount of flow from itself to each of its parent according to parent’s current residual energy to balance relaying node’s energy consumption. However, each cluster has the same cluster size in FBR; thus sensor nodes closer to the sink will die faster.

In this paper, we propose a flow-partitioned unequal clustering routing algorithm (FPUC) aiming to achieve longer network lifetime and coverage lifetime. Network lifetime is defined as the duration from the beginning to the time when any or a given percentage of sensor nodes die. Coverage lifetime measures the period from the network setup time to the time that network coverage drops below a predefined threshold or application’s demanding value [16–18]. Compared with the existing clustering routing algorithms, FPUC mainly has the following differences.

(1) FPUC considers not only sensor node’s residual energy but also sensor node’s overlapping degree in the process of cluster heads selection. The sensor node which has more residual energy and larger overlapping degree has higher probability to be a cluster head.

(2) FPUC computes the competition radius according to sensor node’s distance to the sink and sensor node’s surrounding node density.

(3) After aggregating the data received from CMs, CH sends data to its CMs by the way of partitioning flow. When a CM receives data from its CH, it will transmit the flow to the next hop with the minimum cost. Thus the CH and CMs bear the burden of data relaying together.

The remainder of this paper is organized as follows. Section 2 introduces the related work. The network model and some definitions are described in Section 3. In Section 4, we propose the FPUC algorithm. Section 5 provides performance analysis about FPUC. In Section 6, we present the simulation results to show the validity of FPUC. The last section gives the conclusions.

2. Related Work

A lot of clustering routing algorithms have been proposed in recent years. Low-energy adaptive clustering hierarchy (LEACH) [5] is the first clustering routing protocol proposed. In LEACH, each sensor node has a certain probability to become a cluster head per round and the CHs send the aggregated data to the sink directly. Obviously, LEACH is very simple and can effectively extend the lifetime. However, LEACH does not take the residual energy of sensor node into consideration which will result in that the sensor node with little residual energy may become a cluster head. Moreover, because of transmitting data to the sink via single hop, sensor nodes closer to the sink will die faster, respectively. To solve the problems of LEACH, Younis and Fahmy proposed HEED in which the sensor node having more residual energy has higher probability of becoming cluster head [6]. As a result, HEED can generate a more reasonable network topology. In HEED, it adopts multihop routing. Consequently, sensor nodes closer to the sink would deplete their energy faster. The unbalanced energy consumption among different CHs may cause the network to be partitioned.

For solving hot spots problems in multihop WSNs, Soro and Heinzelman first proposed unequal clustering algorithm [19]. In [19], the network topology is a two-level concentric circle around the sink and the inner ring has smaller cluster size than the outer ring. However, the work in the literature [19] focuses on a heterogeneous network where CHs are deployed at determined locations. In [7], Li et al. proposed an energy-efficient unequal clustering algorithm (EEUC). In EEUC, the cluster closer to the sink has smaller size so that the cluster head can have more energy to relay data. Other unequal clustering algorithms, for example, [20, 21], are, respectively, based on the core idea of EEUC. These existing algorithms compute competition radius only according to sensor node’s distance to the sink so that the cluster in an area where the node density is very high may have too many CMs. As a result, the cluster head may be loaded with too heavy burden and then would die faster. It is much worse in the case that the sensor nodes are deployed nonuniformly.

Research works in the literature [12–15] concentrate on how to switch off as many as possible sensor nodes for saving energy consumption while satisfying application's
requirements for coverage. In [12], the authors proposed a sleep scheduling mechanism that puts sensor nodes into sleeping mode based on coverage information. In [13], the authors proposed a sleep scheduling scheme, where the probability of a sensor node’s entering sleeping state is proportional to its distance to its cluster head. However, achieving the expected coverage ratio does not ensure that it can provide persistent coverage of the monitored area. Obviously, sensor nodes do not equally contribute to the whole network coverage. Comparing with the loss of a sensor node in scarcely populated regions, the death of a sensor node in a densely populated area is not so important. In other words, if some parts of a sensor node’s sensing area are not covered by any other sensor nodes, the loss of this sensor node would leave the network inefficient. Such a sensor node can be called the coverage-critical node. These algorithms select cluster heads according to residual energy or previous activity of the sensor node as cluster head, ignoring sensor node’s overlapping degree. So in these algorithms, the coverage-critical nodes may become cluster heads. Thus, these cluster heads would run out of energy very fast so that the coverage lifetime would be affected. This paper proposes a new cluster head selection algorithm that takes sensor node’s residual energy and overlapping degree into consideration at the same time.

Existing clustering routing algorithms [2–10, 19–21] perform the cluster construction operation periodically to balance the energy consumption between cluster head and cluster members. However, these approaches cannot improve energy balance of relaying nodes significantly. In FBR [16], it assigns the transferred data over multiple paths from sensor nodes to the sink in order to equalize the energy consumption of sensor nodes. But the cluster formation is performed only once in FBR, which will result in that the sensor nodes as cluster heads would deplete energy very fast. And in FBR, each cluster has the same cluster size; thus the energy hole will occur around the sink. In this paper, we propose a novel algorithm where flow-partitioned routing is applied. At the same time, it adopts unequal clustering mechanism that will help avoid the hot spots problem around the sink, thus prolonging network lifetime.

3. Network Model and Definitions

3.1. Network Model. In this paper, we adopt the same network model as in [7, 8, 20], considering $M \times L$ square units, denoted by $S$, in which $N$ homogeneous sensor nodes are randomly deployed. Then the average node density $\rho_0$ is $N/S$. Each sensor node is assigned with a unique ID and has the same initial energy $E^{\text{init}}$. Sensor nodes cannot get location information themselves and all of them will not move after deployment. We assume that the sink is out of the sensing field and has enough energy. The sensing range of a sensor node is $r$. Each sensor node is able to communicate with the sink directly, having a tuned transmission radius, denoted by $d = kR$ ($k = 1, 2, 3, \ldots$) where $R$ is the minimum transmission range and we assume that $R$ is larger than $r$.

The sensor nodes in the range $R$ of sensor node $i$ are called the neighbor nodes of sensor node $i$, denoted by $\text{NG}(i)$. And those sensor nodes in the range from $r$ to $2r$ of sensor node $i$ are called the close nodes of sensor node $i$, denoted by $\text{CN}(i)$. Figure 2 shows an example in which sensor node $i$ has one neighbor node $z$ and two close nodes, $z$ and $o$.

Then, $\text{NG}(i)$ and $\text{CN}(i)$ can be given as follows:

$$\text{NG}(i) = \left\{ j \mid d_{i-j} \leq R \right\},$$

$$\text{CN}(i) = \left\{ j \mid d_{i-j} = r \land d_{i-j} \leq 2r \right\}. \quad (1)$$

The overlapping degree of a sensor node is the ratio of the overlapping area of the sensor node with its close nodes to its sensing area. For example, the overlapping degree of sensor node $i$, denoted by $\text{OPD}(i)$, can be computed as follows:

$$\text{OPD}(i) = \frac{\bigcup_{x \in \text{CN}(i)} S_i \cap S_x}{S_i}. \quad (2)$$

As Figure 3 illustrates, assuming the sensing area of sensor node $i$ is $S_i$ and the area of shadowing portion is $S'$, then the overlapping degree of sensor node $i$ is $S'/S_i$. Obviously, the value of $\text{OPD}(i)$ ranges from 0.0 to 1.0 and the
smaller the OPD(i) is, the more important the sensor node i
to the network is. If OPD(i) = 0, it indicates that there
are no other sensor nodes in the sensing area of sensor node i.
That is to say, the information of the area covered by sensor
node i cannot be got if sensor node i dies. To get OPD(i),
∪ \ x ∈ CN(𝑖)\ S_𝑖 \cap S_x should be calculated first.
First of all, we consider a simple case. As Figure 4 shows,
sensor node i has one close node and the sensing area of i,
denoted by S_i, is \pi r^2. The area of the shadow region part
∪ \ x ∈ CN(𝑖)\ S_𝑖 \cap S_x can be expressed as S_i \cap S_o which can be computed according to
\begin{align}
S_i \cap S_o = 2r^2 \arccos \left( \frac{d_{i-o}}{2r} \right) - \frac{d_{i-o}^2}{2} \sqrt{4r^2 - d_{i-o}^2},
\end{align}
where r is the sensing range and d_{i-o} is the distance from
sensor node i to sensor node o that can be got by the log-
shadowing model [22].
So, when sensor node i has multiple close nodes,
∪ \ x ∈ CN(𝑖)\ S_𝑖 \cap S_x can be calculated according to
\begin{align}
\bigcup x ∈ CN_i S_i \cap S_j = S_i \cap S_j + \bigcup x' ∈ (CN(i) - j) S_i \cap S_{x'},
- \left( S_i \cap S_j \cap \bigcup x' ∈ (CN(i) - j) S_{x'} \right),
\end{align}
where \ j ∈ CN(i).

3.2. Energy Model. This paper only considers the energy
consumed in data transmitting and receiving as the energy
model used in [2, 5–10, 20, 21]. The energy consumed for
data transmitting comes from transmitter electronics and
amplifier, and the energy consumed for data receiving comes
from receiver electronics. Then, when transmitting one l-bit
packet with distance d, the radio expends:
\begin{align}
E_t(i, d) = \begin{cases} 1E_{elec} + 1E_{mp}d^2, & d < d_0 \[2em] 1E_{elec} + 1E_{mp}d^4, & d \geq d_0. \end{cases}
\end{align}
As well, when receiving one l-bit packet, the radio expends:
\begin{align}
E_r(i) = 1E_{elec},
\end{align}
where \ E_{elec} is the transmitting and receiving circuit loss.
The free space (d^2 power loss) is adopted when the distance
between transmitter and receiver is smaller than the thresh-
hold d_o, and the multipath model (d^4 power loss) is adopted
when the distance between transmitter and receiver is larger
than the threshold d_o.

4. The Proposed Algorithm
FPUC consists of two phases: clustering and flow-partitioned
routing. In the clustering phase, it first selects candidate
cluster heads based on sensor node's residual energy ratio.
Then it calculates competition radius according to sensor
node's distance to the sink and surrounding node density.
After that, CHs are generated on the basis of sensor node's
residual energy and overlapping degree. In the routing phase,
CH aggregates data received from CMs and then distributes
the data flow to gateway nodes which will relay the received
data to the next hop with minimum cost.

4.1. Cluster Formation Algorithm. At the time of network
setup, the sink broadcasts a message named "Initial(t_0)" to all
sensor nodes, where t_0 is the time of cluster formation. After
receiving the initial message, each sensor node computes the
distance to the sink according to signal strength and returns
an "Init_Response" message that contains its ID and distance
to the sink.
At the beginning of cluster formation phase, each sen-
 Sor node broadcasts a "Compete_Head_Msg" message that
includes its ID, residual energy, and distance to the sink with
max competition radius \ R_{comp}^max (assuming \ R_{comp}^max > 2r and \ R_{max}^comp > R). Thus, each sensor node, say i, can find its
neighbor nodes NG(i) and close nodes CN(i) according to all
received "Compete_Head_Msg" messages. After that, sensor
node i calculates its residual energy ratio (RER) according to
\begin{align}
RER(i) = \frac{\text{residual energy of sensor node } i}{\text{average residual energy of sensor nodes in } NG(i)} = \frac{E_{\text{residual}}(i)}{E_{\text{residual}}(NG(i))},
\end{align}
where \ E_{\text{residual}}(i) is the residual energy of sensor node i and \ E_{\text{residual}}(NG(i)) is the average residual energy of sensor nodes in
NG(i). If RER(i) >= 1, sensor node i becomes a candidate
cluster head; otherwise, it enters sleeping mode and keeps
sleeping until the cluster head selection ends. If a sensor node
becomes a candidate cluster head, it needs to compute its
competition radius. We set \ R_{\text{comp}}(i) as a function of the distance
to the sink of sensor node i as [7, 20]
\begin{align}
R_{\text{comp}}(i) = \left(1 - \frac{d_{\text{max}} - d_{i-sink}}{d_{\text{max}} - d_{\text{min}}} \right) R_{\text{comp}}^\text{max},
\end{align}
where \ d_{\text{max}} and \ d_{\text{min}} denote the maximum and minimum
distance between sensor nodes and the sink, \ d_{i-sink} is the
distance between sensor node i and the sink, and c is a
constant parameter between 0 and 1. For a candidate cluster
head i, assuming that there are N_i sensor nodes in the range

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{sensor_node_diagram.png}
\caption{Sensor node i with one close node.}
\end{figure}
of its competition radius $R_i^{\text{comp}}$, the surrounding node density of candidate cluster head $i$ can be given by

$$\rho(i) = \frac{N_i}{\pi \times R_i^{\text{comp}} \times R_i^{\text{comp}}}.$$  \hfill (9)

If $\rho(i) - \rho_0 > 0$, the competition radius of candidate cluster head $i$ should be recalculated. Candidate cluster head $i$ first counts the number of sensor nodes in its increasing range $D_0$, $2D_0$, $3D_0$ $\cdots$ $\lambda D_0$ ($\lambda D_0 \leq R_i^{\text{comp}}$); $\lambda$ is the maximum positive integer that ensures that $\lambda D_0$ is not larger than $N_i^{\text{comp}}$, denoted by $N_i(1), N_i(2) \cdots N_i(\lambda)$. Then candidate cluster head $i$ decreases competition radius from $\lambda D_0$ by the step of $D_0$ until the $\rho(i)$ is not larger than $\rho_0$ and adopts this $mD_0$ ($1 \leq m \leq \lambda$) as the final competition radius.

Once finishing computing competition radius, all candidate cluster heads start to compete to be cluster head. In FPUC, it adopts timing broadcast mechanism. The time that a candidate cluster head declares to become a cluster head is defined as follows:

$$t_i = \begin{cases} 
\mu \times \left( \frac{E_{\text{residual}}}{E_{\text{NC}(i)}} \times (1 - \alpha) + (1 - \alpha) \right) (E_{\text{residual}} \geq E_{\text{residual}}^{\text{NC}(i)}) \\
\times (1 - \text{OPD}(i)) \times t_0, (E_{\text{residual}} < E_{\text{residual}}^{\text{NC}(i)}) \\
+\infty, \end{cases}$$

where $NC(i)$ is the candidate cluster heads in the range of $R_i^{\text{comp}}$ of sensor node $i$, $E_{\text{residual}}$ is the residual energy of candidate cluster head $i$, $E_{\text{residual}}^{\text{NC}(i)}$ is the average residual energy of sensor nodes in $NC(i)$, OPD($i$) is the overlapping degree of candidate cluster head $i$, $\mu$ is a random decimal between 0.95 and 1.0 that could reduce the probability of broadcast conflict, and $\alpha$ is a regulator that ranges from 0 to 1. According to (10), the more the residual energy and the larger the overlapping degree the candidate cluster head has, the shorter the waiting time to become a cluster head is. Once a candidate cluster head becomes a cluster head, it will broadcast a message named “Final_Head_Msg” to inform other sensor nodes and if a candidate cluster head receives a “Final_Head_Msg” message, it will quit the cluster head competition.

When the cluster head selection phase ends, ordinary sensor nodes wake up. To inform the ordinary sensor nodes to join cluster, each cluster head broadcasts a message named “Construct_Cluster_Msg.” After that, each ordinary node joins the closest cluster according to signal strength and returns a “Join_Cluster_Msg” to notice the corresponding cluster head.

Algorithm 1 is the pseudocode of cluster head selection algorithm.

4.2 Flow-Partitioned Routing Algorithm. In this phase, each cluster member first sends data to its cluster head. The cluster head aggregates the data gathered from its cluster members into a single packet, which is assumed $L$-bit, and then transmits the data to the sink by the way of flow partitioning, which can significantly balance energy consumption among sensor nodes. Figure 5 illustrates an example; to explain the advantages of flow-partitioned routing more clearly and briefly, we make the following assumptions: (1) sensor node $i$ has a $Y$-bit data packet, denoted by $P_i$, to be conveyed and the residual energy of sensor nodes $z$, $v$, and $g$ is 0.4 J; (2) the energy consumption for transmitting $Y$-bit data from sensor nodes $z$, $v$, and $g$ to the sink is 0.3 J and the energy consumption for receiving $Y$-bit of sensor nodes $z$, $v$, and $g$ is 0.12 J, 0.1 J, and 0.12 J. So if sensor node $i$ sends packet $P$ to the sink via sensor node $v$, sensor node $i$ will run out of its energy. If the data packet $P$ is partitioned into three equal packets, denoted by $p_1$, $p_2$, and $p_3$, each of which is $Y/3$-bit and sensor node $i$ sends $p_1$, $p_2$, and $p_3$ to sensor nodes $z$, $v$, and $g$; thus after the sink receives the three packets from sensor nodes $z$, $v$, and $g$, the residual energy of sensor nodes $z$, $v$, and $g$ is 0.26 J, 0.27 J, and 0.26 J. Obviously, flow-partitioned routing can effectively balance energy consumption among sensor nodes, thus significantly prolonging the network lifetime. The flow-partitioned routing algorithm is composed of two phases: data flow partitioning phase and relaying phase. In the data flow partitioning phase, the cluster head partitions data flow into several small packets and then distributes these packets to its gateway nodes. In the relaying phase, each gateway node transmits received data to the next hop with minimum cost. If a sensor node, say $j$, receives data from a gateway node, sensor node $j$ will send the data to its cluster head.

Figure 6 shows the entire process intuitively. In the data flow partitioning phase, each cluster head, say $j$, first finds its gateway nodes $\text{GN}(j)$ according to

$$\text{GN}(j) = \{ i | i \in \text{Cluster Mem}(j) \land E_i^{\text{residual}} \geq E \land d_{j-sink} \leq d_{j-sink} \}.$$  \hfill (11)
for every node in the network
    broadcast a Compete_Head_Msg
for every node \( j \) in the network
    if receive a Compete_Head_Msg from node \( j \) then
        compute distance between node \( i \) and node \( j \)
    end if
for every node \( i \) in the network
    find NG\((i)\) and CN\((i)\)
    compute OPD\((i)\) according to (2), (3), (4)
    compute RER\((i)\) according to (7)
    if \( RER(i) >= 1 \) then
        beCandidateClusterHead ← TRUE
        compute \( R_{comp} \) of node \( i \) according to (8), (9)
    else
        sleep
    end if
for every CandidateClusterHead in the network
    compute \( t_i \) according to (10)
while (the timer \( t_i \) is not expired)
    if \( (CurrentTime < t_i) \)
        if receive a Final_Head_Msg
            give up the competition
        end if
    elseif \( (CurrentTime == t_i) \)
        beClusterHead ← TRUE
        broadcast a Final_Head_Msg
    end if
end while

Algorithm 1: The pseudocode of cluster head selection algorithm.

Assuming sensor node \( j \) has \( H \)-bit data to be transmitted, the amount of data to be distributed to each of its gateway nodes, say \( i \), is given as follows:

\[
h_j = \frac{E_{\text{residual}}^j}{\sum_{v \in \text{GN}(j)} E_{\text{residual}}^v} H, \quad i \in \text{GN}(j). \tag{12}
\]

When a gateway node, say \( i \), receives data sent by its cluster head, it needs to find the next hop to forward data. In order to reduce energy consumption for communication, a distributed strategy is adopted where gateway nodes choose an optimal path for transmission. At the beginning, gateway node \( i \) should decide whether to send data to the sink directly or broadcast a message which includes its ID, its cluster head ID, and its distance to the sink depending on whether its distance to the sink \( d_{i-sink} \) is larger than the threshold \( T_D \). When a sensor node, say \( j \), receives the message from gateway node \( i \), it will give gateway node \( i \) a response message that contains its ID, its residual energy \( E_{\text{residual}}^i \), its distance to the sink \( d_{j-sink} \), its cluster head's residual energy \( E_{\text{CH}} \), and its cluster head's distance to the sink \( d_{\text{CH-sink}} \) if it is not in the same cluster as gateway node \( i \) and it is closer to the sink than gateway node \( i \). After receiving the response message from sensor node \( j \), gateway node \( i \) computes the distance to sensor node \( jd_{i-j} \) and adds the information of sensor node \( j \) into the candidate relaying nodes table CRN\((i)\). At last, gateway node \( i \) will select the sensor node with minimum cost in CRN\((i)\) as the next hop. The cost in this paper is defined as follows:

\[
\text{cost}(i, j) = \theta \frac{E_{\text{init}}^i}{E_{\text{residual}}^i} + \eta \frac{E_{\text{init}}^j}{E_{\text{residual}}^j} + \omega \frac{d_{i-j}^2 + d_{i-\text{CH}}^2 + d_{\text{CH-sink}}^2}{d_{i-sink}^2}, \tag{13}
\]

\[
\theta + \eta + \omega = 1,
\]

where \( \theta, \eta, \) and \( \omega \) are parameter factors. The cost function is mainly based on the following considerations.

1. The first term in the cost function aims to select the sensor node with more residual energy as relaying node to forward data. Because data transmission will consume energy, the residual energy of a sensor node is a very important factor.
2. The second term in the cost function takes the residual energy of candidate relaying node's cluster head into consideration because when a relaying node receives data, it will send the data to its cluster head.
3. The third term in the cost function aims to select the route that uses minimum energy to transmit data from sensor node \( i \) to the sink.
In the data flow partitioning phase, it may occur that some cluster heads, say \( j \), may fail to get their gateway nodes, called “Gateway Hole.” As Figure 7 shows, each of the cluster members that are closer to the sink of cluster head \( j \) does not have more residual energy than the cluster’s average residual energy. If the problem arises, cluster head \( j \) will broadcast a message that contains its distance to the sink with \( \delta R_{\text{comp}} \) (\( \delta \) is the minimum positive integer that ensures that \( \text{NCH}(j) \) is not empty). When the cluster head that is closer to the sink than cluster head \( j \), say \( i \), receives the message, it will return a message that contains its residual energy. After receiving these messages, cluster head \( j \) will set up a neighbor cluster heads table \( \text{NCH}(j) \) and add the information of cluster head \( i \) into \( \text{NCH}(j) \). The amount of data distributed to each cluster head in \( \text{NCH}(j) \) is given as follows:

\[
h_i = \frac{E_{\text{residual}}^i}{\sum_{v \in \text{NCH}(j)} E_{\text{residual}}^v} H, \quad i \in \text{NCH}(j).
\]  

5. Performance Analysis

**Lemma 1.** In the cluster formation phase, the message complexity is \( O(N) \), where \( N \) is the total number of sensor nodes in the network.

**Proof.** At the beginning of the cluster selection phase, each of the sensor nodes broadcasts a “Compete_Head_Msg” message, so the total number of “Compete_Head_Msg” messages is \( N \). Assuming that in the cluster head selection phase, \( Q \) cluster heads are generated. Thus these \( Q \) cluster heads will broadcast \( Q \) “Final_Head_Msg” messages. To construct cluster, each cluster head should broadcast a “Construct_Cluster_Msg” message and each ordinary sensor node should send a “Join_Cluster_Msg” message. The number of these two types of messages is \( N \). So the total messages add up to \( N + Q + N = 2N + Q \).

Thus the message complexity is \( O(N) \). \( \square \)

**Lemma 2.** In cluster formation phase, the upper bound for the number of cluster heads is \( 2S/\sqrt{3}((1 - c)R_{\text{comp max}})^2 \) and the lower bound is \( 2S/3\sqrt{3}(R_{\text{comp max}})^2 \).

**Proof.** For each cluster head, there are no other cluster heads in the range of its competition radius. In [22], it has been proved that the maximum overlapping angle between three adjacent disconnected sensor nodes is \( \pi/3 \). As Figure 8 shows, the number of cluster heads in the network reaches the maximum value. It can be computed that the area of a cluster is \( \sqrt{3}(R_{\text{comp}})^2/2 \). Figure 9 shows an opposite case. The area of a cluster reaches the maximum value \( 3\sqrt{3}(R_{\text{comp}})^2/2 \), which has been proved in [23]. It can be easily seen that \( R_{\text{comp}} \) ranges from \((1 - c)R_{\text{max}}\) to \( R_{\text{comp}} \) therefore, the upper bound for the number of cluster heads is \( 2S/\sqrt{3}((1 - c)R_{\text{comp max}})^2 \) and the lower bound is \( 2S/3\sqrt{3}(R_{\text{comp max}})^2 \). \( \square \)

**Lemma 3.** The total number of messages in the cluster formation phase is smaller than \( 2N + [2S/\sqrt{3}((1 - c)R_{\text{comp max}})^2] \).
Proof. It can be concluded from Lemma 1 that the total number of messages in the cluster formation phase is $2N + Q$. Obviously, we have $Q < \lceil 2S / \sqrt{3}((1 - c)R_{\text{comp}}^\text{max})^2 \rceil$; thus $2N + Q < 2N + \lceil 2S / \sqrt{3}((1 - c)R_{\text{comp}}^\text{max})^2 \rceil$.

6. Simulation Results

In this section, we provide simulation results to validate the effectiveness of FPUC and compare the performance with existing algorithms: LEACH [5], EEUC [7], and FBR [16]. Firstly, we show two performance metrics of the four algorithms: network lifetime and coverage lifetime. Then we analyze the energy balance and overall energy consumption of each algorithm. Besides, to observe the effects of parameters on performance, we also change the network parameters to see the difference. At last, we compare the performance of the four algorithms under uniform and nonuniform deployment situations. The network environment and experimental default parameters are given in Table 1.

6.1. Comparison of Network Lifetime. In this paper, we use two metrics: network lifetime and coverage lifetime to evaluate the performance of LEACH [5], EEUC [7], FBR [16], and FPUC. Network lifetime is defined as the duration from the beginning to the time when any or a given percentage of sensor nodes die. Coverage lifetime measures the period from the network setup time to the time that network
coverage drops below a predefined threshold or application’s requirement.

Figure 10 compares the network lifetime of FPUC with that of LEACH, EEUC, and FBR. From the results shown in the picture, we can see that FBR outperforms other approaches and FPUC yields a much longer network lifetime than FBR.

Figure 11 compares the coverage lifetime of FPUC along with LEACH, EEUC, and FBR. Respectively, FPUC can achieve better coverage lifetime than the other three protocols and when the first sensor node runs out of energy, the network coverage ratio is still kept at 100%.

6.2. Energy Consumption. For wireless sensor networks, the algorithms that aim to improve the lifetime need to have the following two features: (1) minimum total energy usage; (2) balanced energy consumption. Figure 12 shows the energy consumption over time of the four algorithms. Figure 13 represents each sensor node’s residual energy at the 100th round of LEACH, EEUC, FBR, and FPUC.

We can have the following conclusions from Figures 12 and 13. The energy consumption in each round of LEACH and EEUC is much more than that of FBR. And FPUC consumes relatively less energy than FBR. In Figure 13, we can clearly know each sensor node’s residual energy at the 100th round of the four algorithms. Figure 13(a) is the distribution of each sensor node’s residual energy of LEACH. It can be easily seen that the sensor nodes closer to the sink have much more energy than those sensor nodes far away from the sink. Figure 13(b) is the distribution of each sensor node’s residual energy of EEUC. Obviously, the sensor nodes far away from the sink have much more energy than those sensor nodes closer to the sink. Figure 13(c) is the distribution of each sensor node’s residual energy of FBR. Respectively, the energy consumption of FBR is relatively more balanced than LEACH and EEUC. Figure 13(d) is the distribution of each sensor node’s residual energy of FPUC. Compared to LEACH, EEUC, and FBR, the total energy consumption among all sensor nodes of FPUC is more balanced. However, the sensor nodes closer to the sink still consume more energy.

The reasons for the above results can be summarized as follows. In LEACH, the data is transmitted to the sink directly; that is to say, the farther the sensor node is, the more energy the sensor node will consume when forwarding data. Thus, the sensor nodes far away from the sink will consume more energy. In EEUC, it adopts multihop data transmission,
so the sensor nodes closer to the sink will be loaded with more data forwarding traffic, thus consuming more energy. In FBR, because the flow-balanced routing is used, the energy consumption is balanced in a degree. However, because only the cluster heads convey data and the cluster formation is performed only once, the cluster heads will run out of energy faster. In FPUC, the cluster heads and cluster members share the burden of data traffic at the same time and unequal clustering is adopted, thus leading to more balanced energy consumption.

Figure 14 shows the distribution of dead sensor nodes when 10% sensor nodes die. We can see that the distribution of FPUC is more uniform in comparison with the other three algorithms.

6.3. The Effects of Network Parameters on Performance

6.3.1. Varying the Number of Sensor Nodes. First we evaluate how the number of sensor nodes affects the performance. From Figure 15, we can draw the conclusion that the number of sensor nodes has a little effect on the performance of LEACH. When the number of sensor nodes increases, EEUC, FBR, and FPUC can achieve longer network lifetime in some degree. In EEUC, increasing the number of sensor nodes can contribute to select a more optimal relaying node when forwarding data. In FBR and FPUC, there will be more sensor nodes to share the burden of data transmission if the number of sensor nodes increases and the energy consumption will be more balanced.

6.3.2. Varying the Value of Parameter $\alpha$. Figure 16 shows how the parameter $\alpha$ affects the network lifetime of FPUC and Figure 17 shows the effects of parameter $\alpha$ on the coverage lifetime. It can be seen that the performance is the best when $\alpha$ is 0.5.

6.3.3. Varying the Node Distribution. From the results in Figure 18, we can see that the network lifetime of LEACH, EEUC, and FBR under the uniform distribution is longer than that under the nonuniform distribution. But the distribution has a little effect on the performance of FPUC because it can also generate a reasonable topology under the nonuniform distribution.
7. Summary

In this paper, we have proposed a new flow-partitioned unequal clustering routing algorithm. In the clustering formation phase, it first computes the competition radius according to the node density and the distance from a sensor node to the sink. Then it selects the sensor nodes with more residual energy and larger overlapping degree as cluster heads. In the data transmitting phase, flow-partitioned routing is adopted, where cluster heads and cluster members share the burden of data forwarding together. Comprehensive simulation results demonstrate that FPUC can significantly improve network lifetime and coverage lifetime comparing with LEACH, EEUC, and FBR.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61303204, U1333113), the Sichuan Province Science and Technology Support Plan (2012GZX0088-1, 2011GZ0188), and the Scientific Research Fund of Sichuan Normal University (13KYL06).

References

[1] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” Ad Hoc Networks, vol. 3, no. 3, pp. 325–349, 2005.

[2] T. Liu, Q. R. Li, and P. Liang, “An energy-balancing clustering approach for gradient-based routing in wireless sensor networks,” Computer Communications, vol. 35, no. 17, pp. 2150–2161, 2012.

[3] A. F. Liu, Z. H. Liu, M. Nurudeen, X. Jin, and Z. G. Chen, “An elaborate chronological and spatial analysis of energy hole for
wireless sensor networks,” *Computer Standards & Interfaces*, vol. 35, no. 1, pp. 132–149, 2013.

[4] A. F. Liu, X. Jin, G. H. Cui, and Z. G. Chen, “Deployment guidelines for achieving maximum lifetime and avoiding energy holes in sensor network,” *Information Sciences*, vol. 230, no. 1, pp. 197–226, 2013.

[5] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2012.

[6] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[7] C. Li, M. Ye, G. Chen, and J. Wu, “An energy-efficient unequal clustering mechanism for wireless sensor networks,” in *Proceedings of the 2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS ’05)*, pp. 597–604, Washington, DC, USA, November 2005.

[8] H. Bagci and A. Yazici, “An energy aware fuzzy approach to unequal clustering in wireless sensor networks,” *Applied Soft Computing*, vol. 13, no. 4, pp. 1741–1749, 2013.

[9] D. Kumar, T. C. Aseri, and R. B. Patel, “EEHC: energy efficient heterogeneous clustered scheme for wireless sensor networks,” *Computer Communications*, vol. 32, no. 4, pp. 662–667, 2009.

[10] D. W. Gong, Y. Y. Yang, and Z. X. Pan, “Energy-efficient clustering in lossy wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 73, no. 9, pp. 1323–1336, 2013.

[11] W. K. Lai, C. S. Fan, and L. Y. Lin, “Arranging cluster sizes and transmission ranges for wireless sensor networks,” *Information Sciences*, vol. 183, no. 1, pp. 117–131, 2012.

[12] X. Wang, G. Xing, Y. Zhang, and C. Lu, “Integrated coverage and connectivity configuration in wireless networks,” *ACM Transactions on Sensor Networks*, vol. 1, no. 1, pp. 36–72, 2005.

[13] J. Deng, Y. S. Han, W. B. Heinzelman, and P. K. Varshney, “Scheduling sleeping nodes in high density cluster-based sensor networks,” *Mobile Networks and Applications*, vol. 10, no. 6, pp. 825–835, 2005.

[14] F. Shen, M.-T. Sun, C. Liu, and A. Salazar, “Coverage-aware sleep scheduling for cluster-based sensor networks,” in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’09)*, pp. 1–6, April 2009.

[15] N. Xu, A. Huang, T.-W. Hou, and H.-H. Chen, “Coverage and connectivity guaranteed topology control algorithm for cluster-based wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 12, no. 1, pp. 23–32, 2012.

[16] Y. L. Tao, Y. B. Zhang, and Y. S. Ji, “Flow-balanced routing for multi-hop clustered wireless sensor networks,” *Ad Hoc Networks*, vol. 11, no. 1, pp. 541–554, 2013.

[17] I. Dietrich and F. Dressler, “On the lifetime of wireless sensor networks,” *ACM Transactions on Sensor Networks*, vol. 5, no. 1, article 5, 2009.

[18] A. A. Abbasi and M. Younis, “A survey on clustering algorithms for wireless sensor networks,” *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[19] S. Soro and W. B. Heinzelman, “Prolonging the lifetime of wireless sensor networks via unequal clustering,” in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS ’05)*, April 2005.