information

MDPI

Article

Multi-Path Data Distribution Mechanism Based on RPL for Energy Consumption and Time Delay

Licai Zhu [1,2,*], Ruchuan Wang [2] and Hao Yang [1]

[1] School of Information Engineering, Yancheng Teachers University, Yancheng 224002, China; yctc_cai@126.com

[2] School of Computer Science & Technology, Nanjing University of Posts and Telecommunications, Nanjing 210042, China; wangrc@njupt.edu.cn

* Correspondence: 2013010220@njupt.edu.cn; Tel.: +86-158-6198-7987

Received: 4 September 2017; Accepted: 2 October 2017; Published: 9 October 2017

Abstract: The RPL (Routing Protocol for LLN) protocol is a routing protocol for low power and lossy networks. In such a network, energy is a very scarce resource, so many studies are focused on minimizing global energy consumption. End-to-end latency is another important performance indicator of the network, but existing research tends to focus more on energy consumption and ignore the end-to-end delay of data transmission. In this paper, we propose a kind of energy equalization routing protocol to maximize the surviving time of the restricted nodes so that the energy consumed by each node is close to each other. At the same time, a multi-path forwarding route is proposed based on the cache utilization. The data is sent to the sink node through different parent nodes at a certain probability, not only by selecting the preferred parent node, thus avoiding buffer overflow and reducing end-to-end delay. Finally, the two algorithms are combined to accommodate different application scenarios. The experimental results show that the proposed three improved schemes improve the reliability of the routing, extend the lifetime of the network, reduce the end-to-end delay, and reduce the number of DAG reconfigurations.

Keywords: data distribution; multi-path; RPL; wireless sensor network

1. Introduction

In recent years, wireless sensor networks in smart homes [1], environmental surveillance [2], intelligent buildings [3] and many other areas have been widely used. The topology of this network is instable, due to its intrinsic character and interference from surroundings, thus its link quality will sometimes change. More and more attention has been paid to how to effectively distribute data, especially for low power and lossy networks (LLN). IETF's ROLL (Routing over Lossy and Low-power Networks) group has proposed RPL [4] (routing protocol for low-power and lossy networks) to availably manage data traffic.

As is well-known, lifetime and end-to-end delay are two key features of network quality. To ensure that information is effective and real-time, data should be sent to the sink node as soon as possible. However, partial nodes may have to undertake many communication tasks, which leads to their energy being worn out, therefore cutting down the lifetime of the whole network. On the other hand, we need to transmit abundant traffic and node buffers will probably overflow. This increases the delay of network transmission, meaning that it is contradictory to prolonging the lifetime and reducing delays. For the former, we should balance the energy costs of all nodes. In practice, if some nodes are always selected to transmit data, it will result in an energy imbalance, and consequentially shorten the network lifetime. These nodes are the so-called bottleneck nodes. In other words, we should design a suitable transmission strategy to balance energy consumption to avoid the appearance of bottleneck nodes. For the latter, we should select a good quality link to reduce transmission delay. The overflow is

a vital factor leading to data packet dropout and will increase end-to-end delay, since a node can only receive data but not store it when its buffer is full. In this case, it will become a bottleneck node even if it maintains a lot of energy. Worse, it will waste its descendant node's energy. Hence, we should balance node caches to decrease the delay of transmission.

RPL allows one node to have multiple parent nodes, but only one preferred parent node works. It transmits its traffic through this preferred parent node and the other parent nodes are just backups. This makes the preferred parent node easily become the bottleneck node, and the energy consumption of these nodes is significantly accelerated. In other words, their energy and cache imbalance will seriously affect network performance. Therefore, avoiding the bottleneck node is an urgent problem to be solved. We need to design a reasonable data transmission method during communication.

Researchers have considered adding more sink nodes [5] or multiple paths for packet forwarding. The former increases the costs of deployment and the complexity of control. This method is particularly unsuitable for an environment with a lack of infrastructure. For the latter, it can achieve energy balance to some extent, however, its performance depends on the metric used. For example, some metrics may be able to balance traffic, but this also leads to an increase of controlled information and consequently cuts down the lifetime of the network.

This paper proposes an algorithm of adaptive multipath traffic loading based on RPL. Its basic idea is to adaptively distribute traffic through a multipath according to network's real situation. It can avoid the appearance of bottleneck nodes by balancing the nodes' energy, which aims to ensure that the network energy is balanced and to reduce the delay from end to end. We first established a more realistic node energy model considering the node's energy consumption for receiving, sending and calculating. At the same time, a kind of energy dispersion measure is proposed, which can effectively judge the energy balance of nodes. Then, we propose a quick algorithm to obtain the optimal distribution based on the measurement. On the other hand, in order to reduce the end to end delay, we abstract the data forwarding model to M/M/1/S/FCFS (First Come, First Served), and calculate the average waiting time, then present the traffic distribution algorithm based on the greedy idea.

The remainder of this paper is organized as follows. Section 2 introduces the basic working principle of RPL, the related work on multipath routing, energy-aware routing, and end-to-end delay. A new adaptive multipath data distribution algorithm based on RPL is put forward in Section 3. Section 4 evaluates the performance of the algorithm on the real test bed, including the network load distribution, network lifetime, energy consumption, end-to-end delay and other network performance at different network sizes and nodes with different cache sizes. Finally, a conclusion together with an outlook is addressed in Section 5.

2. Related Work

2.1. RPL: Routing Protocol for Low Power and Lossy Networks

Recently, researchers on routing protocol have mainly concentrated on two aspects: hierarchical models and flat models. The hierarchical model is to construct the network into multiple clusters, with every node in the cluster in communication with each other. In this case, the implementation of load-balanced routing depends on the structure of these clusters. This model requires the nodes in the clusters to communicate with each other directly, thus the cluster communication distance is one-hop and is not suitable for a large-scale network. In the flat model, the data is transmitted through multi-hop, and it is suitable for large-scale network deployment, but it cannot communicate with an IP network directly and needs gateways to convert protocols.

The IETF ROLL working group designed the wireless sensor network routing protocol, RPL. It is a kind of distance vector routing protocol which is suitable for low power and lossy networks. It constructs DODAG (Destination-Oriented Directed Acyclic Graph) through one or several routing

metrics to perform data communication. In the process of building and maintaining a DODAG, four RPL messages are needed:

- DIO (DODAG information object): This message is used to build DODAG, which is broadcasted by the root node initially.
- DIS (destination advertisement solicitation): When a node wants to join DODAG, but it does not receive DIO, the node can actively send a DIS message to apply to join DODAG.
- DAO (destination advertisement object): Used to produce reverse routing information. Apart from root nodes, every other node can send DAO.
- DAO-ACK (DAO acknowledgement): Used to confirm DAO.

According to RPL protocol, a network can contain several DODAGs. During the process of constructing a DODAG, the root node broadcasts the DIO message at first including the instance ID, DODAG ID, increasing version number, the rank value and the objective function. After listening to the DIO message, the neighbor of the root node decides whether to join the DODAG. If they join the DODAG, they will calculate their own rank values independently, and send information about the final version number and DODAG identification to neighbors. The same procedure will continue until the leaf nodes.

RPL is designed for LLNs and performs routing in a distributed way. However, load balance is missing in the original RPL. For fault tolerance, each node needs to save a list containing multiple parent node information. Nevertheless, only the preferred parent, which is selected by routing metrics or constraints, is responsible for the communication task, while other parent nodes act as backups. In the real network, some nodes have more neighbors than others and thus spend more energy. If these nodes are chosen to be the preferred parents, a network hole will appear because of their energy exhaustion and the network will disconnect.

2.2. Multipath Routing

Load imbalance directly affects both the performance of the network and its lifetime. Multiple sinks are not suitable in general, although it could alleviate the situation a little. In this case, a multipath could be considered. If we only consider one sink, there are two methods at present. First, minimize the overall energy consumption, for example, using the ETX (Expected Transmission Count) metric to choose a high-efficiency energy link. However, only considering the paths between the nodes and the sink cannot represent the true conditions of communication in practice [6], since this will make some nodes have a large number of children and lead them to run out of energy quickly. Second, nodes with more energy are applied to forward traffic. Nevertheless, these nodes have many poor links of their own. Sustaining transmission through the low-quality link would make the nodes lose many packets.

Multipaths have been widely used to balance loads and improve service quality [7,8]. In [9], the authors chose the preferred parent using the hop count and proposed the protocol DMR (DAG-based Multipath Routing-protocol) for the mobile sensor network. When all the parents of a node are out of work, it is able to construct paths through its siblings. There are two disadvantages to this method. First, it is obviously unreasonable to think that the link with fewer nodes can provide higher quality. Second, frequent changes of topology are bound to result in an increase of both the control packets and the extra consumption of the nodes, and shorten the network lifetime. In [10], a multi-path routing protocol was designed in consideration of node residual energy and ETX. However, it does not consider the energy consumption of the whole network. In [11], the authors propose an energy equalization routing protocol-EBMR, and applied it to the wireless sensor network. However, EBMR uses client/server software architecture to calculate the optimized path in a centralized manner, and to collect the entire wireless topology through the base station. This method is difficult to implement in practice, and the author did not give a detailed design of the algorithm. The literature [12], uses an REER (Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks) metric which combines residual energy, buffer size, and SNR (Signal-to-Noise Ratio) weighted function.

However, this combination implicitly assumes that the energy consumption is linearly dependent on these factors. Reference [13] proposes a multipath equalization algorithm based on a cache of each node. It determines the transmission time of the DIO message according to the size of the remaining cache and then forms a DIO message queue. To achieve the purpose of balancing the load, several parent nodes are selected to send the traffic. However, the author does not consider energy spends and thus it ca not increase the whole network lifetime. In [14], the authors present a HeLD (heuristic load distribution) algorithm based on the multipath extension of standard RPL, trying to achieve a balanced traffic load while maximizing total throughput in the network lifetime. Reference [15–17] achieves a multipath by modifying the underlying layer or controlling parameters. In [15], the authors modify cluster heads in the MAC (Media Access Control) layer and expand RPL through a defining opportunity transmission scheme. This scheme only considers delay but does not guarantee traffic balance, thus it does not prolong the network lifetime. Reference [16] proposes an opportunity-routing ORW (Opportunistic Routing in Wireless sensor networks). In this routing, a packet can reach its destination by using several duty-cycled wakeups. Each node selects a number of potential parents for forwarding, and then uses the coordination algorithm to select a single node as a unique forwarder. However, it needs to modify the MAC layer and this is difficult to achieve. The literature [17] first formalizes the maximum survival time of the network, and then presents an optimal load balancing solution. Load balancing is achieved by transmitting power control to maximize the network lifetime. Although their approach is designed for converged transmissions, they assume that the link quality is constant and homogeneous, which is rare in actual deployments.

In [18], the authors find bottleneck nodes by analyzing the residual time of nodes. Depending on their residual time, a different proportion of the traffic can be distributed for traffic balance. For the nodes with much residual energy, their cache is not sufficient, because they undertake more traffic. This will lead to frequent data loss, frequent retransmission of data and extra energy consumption. This method cannot achieve the purpose of extending the network survival time. Therefore, a good routing algorithm should take into account residual energy and cache size to get a dynamic balance.

2.3. Energy-Aware Routing

Energy assumption is the key issue for a wireless sense network. Reference [6] focuses on minimizing the average energy consumption. The metric ETX is taken into account and energy-efficient routes are constructed based on the link reliability. A small number of nodes with a high ETX or close to the border routers may have to forward most of the traffic. As shown in Figure 1, a DAG is built on ETX. *D* can select *C* or *B* as the next hop. *C* should be preferred because it provides the lowest cumulative ETX to the sink. However, if all nodes produce the same amount of traffic, *B* should be the best choice to balance energy consumption. In [19], the authors divided the traffic into fixed rate and random rate, and transferred the routing problem into a linear planning problem in order to maximize network lifetime. It supposes that all nodes are fixed and the topology changes are slow. Furthermore, it only presents a centralized algorithm and could not support distribution situations. Reference [20] divides energy sources into three kinds—mains-powered, primary batteries and energy scavengers—and presents a calculation standard for every means of power supply. By dynamically adjusting the transmission power of the sensor, the time-insensitive packet is transmitted at a lower power to reduce the energy consumption, and real-time packets are transmitted at a high power. Although this solution is energy efficient for a single node, it does not provide a global solution and does not maximize the network lifetime.

A novel wave routing framework is proposed in [21]. It supports a multi-source, multi-sink node routing scheme for wireless sensor networks. It adopts a distributed, scalable, class potential, field evaluation algorithm. This method constructs multiple paths and balances the load according to the reciprocal of the cumulative path cost. The authors use residual energy as a routing metric to optimize energy consumption. However, this method can only reduce the path cumulative energy, and does not focus on heavy load nodes. In addition, the concept of this potential field is similar to the rank of

RPL. The application of multi-path construction has some limitations. In [22], the authors analyze the reliability measurement and energy measurement. They think that if only the reliability measurement is considered, the distribution of energy between the nodes will not be balanced; if only the energy measurement is considered, some nodes will suffer a higher rate of package loss. They put forward a solution for the linear combination of the nodes' residual energy and ETX. However, this weight is not directly related to the real lifetime of a node. In [23], three RPL multipath schemes are proposed, namely, FLR (fast local repair), ELB (energy load balancing), and their combination FLR-ELB. Then authors apply them to the IPv6 (Internet Protocol Version 6) communication stack for the internet of things. In [24], the authors propose a neighbor node disjoint multipath (NDM) solution, which is proved more efficient when the intermediate node or link fails. In [25], the authors propose cooperative RPL, which creates different instances based on different sensing tasks. Nodes in different instances are responsible for forwarding different data. In this way, energy consumption is reduced compared to standard RPL.

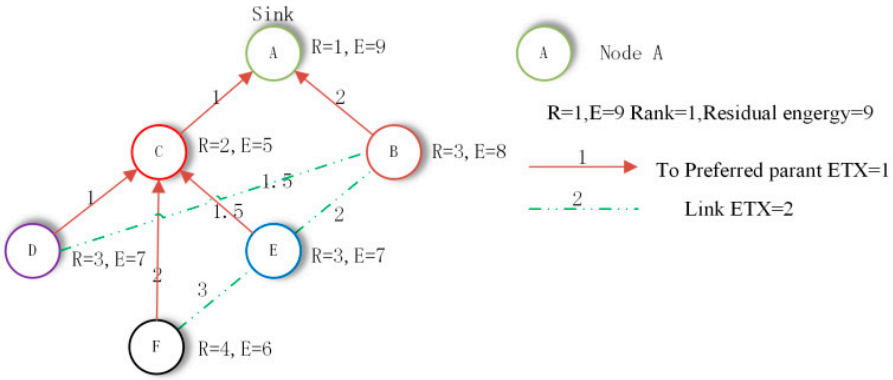

Figure 1. Topological structure based on ETX.

Figure 2 constructs DAG based on the residual energy. F chooses E or C as a next hop. Even though the link quality of E is very low (ETX = 3), it will be chosen as the parent owing to its residual energy being greater. This will increase the number of data package retransmissions of F. According to the figure, C is a more appropriate choice.

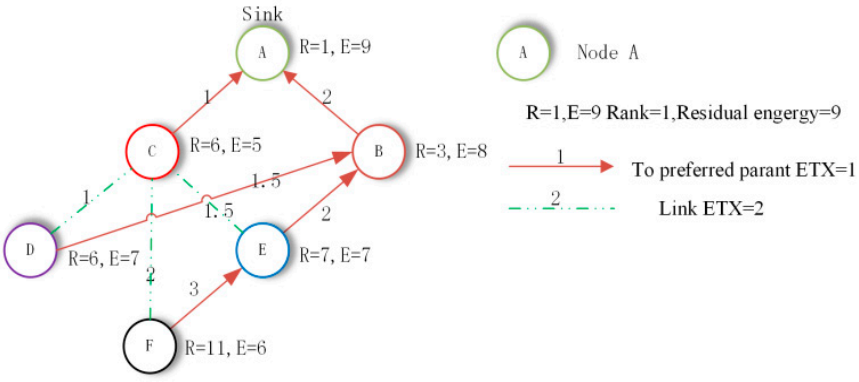

Figure 2. Network topology constructed according to the residual energy.

2.4. End-to-End Delay

In many applications, energy effect and end-to-end delay are related. In order to increase the lifetime of a battery-provided sensor, it should be close the wireless transceiver; otherwise, the data cannot arrive at the objective node in time, which can increase the end-to-end time. To increase the network lifetime, we should use a high-efficiency duty-cycled factor, which is bound to influence the

system delay. High-efficiency means an increase in communication time. It will increase the single-hop delay. Under the circumstance of multi-hop, the end-to-end delay will be increased.

In [26], the authors study the problem of network delay with low duty cycle and compare it with the RPL based on ETX. At the same time, it expands the duty cycle of ContikiMAC (Media Access Control of Contiki) to support different sleep modes. However, it only uses a simple delay metric, but does not consider the quality of the link and failing to build an efficient energy metric mode. It fails to improve the remaining performance of the network and is not verified in a real environment. In [27], the authors use the residual energy and transmission delay as the basis of the next hop choice. In some ways, it considers the network lifetime and end-to-end delay and optimizes the network by ant colony optimization. However, the author did not consider the multipath support. Some nodes still may fail as a result of too much energy consumption. In [28], the authors use fuzzy logic to design objective function and consider node and link measurement, namely, end-to-end delay, hops, ETX and LQI (Link Quality Indicator). This algorithm supports the quality of service in a static and dynamic network environment, and improves the reliability and average delay of the network. The author ignores the node energy consumption, not considering the network lifetime. In [29], opportunistic routing is proposed, which will probably obtain better load balance. The data forwarding decision depends on the receiver rodes, rather than the sender nodes.

3. Adaptive Multipath Traffic Loading Based on RPL, AMTL-RPL

This section present an adaptive multipath traffic loading distribution method based on RPL. We realize the entire network's energy balance and reduce the end-to-end delay by evaluating the energy consumption and cache utilization of bottleneck nodes in the network.

3.1. Energy Balance Based on the RPL

The first concern is the energy balance of the network. According to the idea of multipath transmission, in order to ensure the residual energy balance of the subsequent nodes, the sending node will transmit its data through multiple paths. This method will lead to multiple bottleneck nodes in the network, and makes it necessary to maintain the information of multiple bottleneck nodes. This section describes how to calculate the residual energy of the bottleneck node and how to allocate its transmission traffic. Specifically, we first analyze the basic energy consumption of the node, and then according to the Pareto evaluation model [30] to provide a measurement of the node's energy distribution. Afterwards, we analyze the node's energy consumption situation based on multipath distribution and finally propose the best plan for traffic distribution.

3.1.1. Node Energy Consumption Model

Let us take the network topology of Figure 3 as an example. Node A is the sending node; B1, B2 and B3 are its parent nodes; and C1 and C2 are bottleneck nodes.

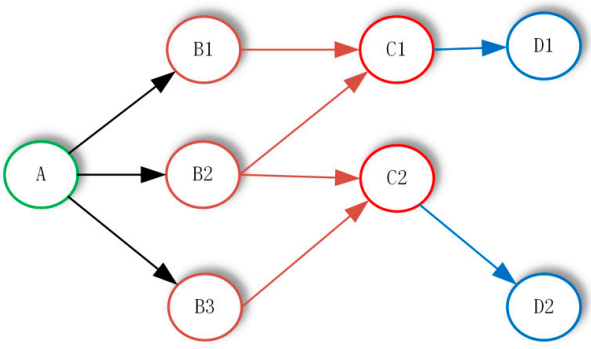

Figure 3. The node's energy consumption.

The node's consumption can be classified as three parts, the receiving consumption for new data, the calculation consumption and the sending consumption. We take node C1 as an example to calculate the three different kinds of energy consumption. The notations used in the article are shown in Table 1.

Step 1: Node C1 receives the data which the descendant node sends; its energy consumption mainly comes from the consumption in receiving data:

$$E_{rce}(c_1) = \sum_{Parents(b)=C1} \frac{Tat(b,c_1)}{PRR(b,c_1)} \times \frac{E_r}{D_r} \quad (1)$$

Step 2: Node C1 handles the cache queue; we can calculate its energy consumption by:

$$E_{cmp}(c_1) = N_{task}(Buffer\_Use(c_1)) \times E_c \quad (2)$$

Step 3: Node C1 sends data to its descendant nodes after finishing to process the cache queue, which energy consumption is:

$$E_{tat}(c_1) = \frac{Tat(c_1,d)}{PRR(c_1,d)} \times \frac{E_t}{D_d} \quad (3)$$

Thus, node C1's residual energy is:

$$E_{rse}(c_1) = E_{crt}(c_1) - E_{tat}(c_1) - E_{cmp}(c_1) - E_{rce}(c_1) \quad (4)$$

Here, $E_{rse(c_1)}$ represents the residual energy after handling a transmission task or the energy used to clear up the cache. This metric can affect the route choice of bottleneck nodes.

Table 1. Notation used in the article.

Notation	Meaning	Notation	Meaning
$E_{tat}(x)$	Sending energy consumption of x	T_r	Receive time
$E_{rce}(x)$	Receiving energy consumption of x	E_{crt}	Current energy
$E_{cmp}(x)$	Calculating energy consumption of x	E_{rse}	Residual energy
E_t	Sending energy consumption per second	$Tat(x,y)$	Data transfer from x to y
E_r	Receiving energy consumption per second	$Buffer\_Use(x)$	Node x cache usage
E_c	Energy consumption per instruction cycle	N_{task}	Task processing instruction cycle
D_d	Sending rate	D_p	Processing rate
D_r	Receiving rate	$PRR(x,y)$	x received a number of ACK packets
S_r	Receive data	T_p	Processing time of each instruction

3.1.2. The Metric Definition of Energy Balance

The amount of data transmission plays an important role for the whole network's transmission rate. Even given global information, choosing an optimized energy balance algorithm is still challenging. This paper uses a multipath to balance the overload of the network's traffic.

According to RPL protocol, once DODAG has been constructed, all the traffic would transmit through one path until the network topology changes. To ensure the maximum network lifetime, we need to improve DODAG, which means making a multi-path available. One node could distribute traffic to multiple parent nodes instead of the preferred one. This ensures that all of parents can be energy balanced. We distribute data among different parent nodes to make sure that the overload of the residual energy of the subsequent bottleneck nodes becomes balanced. In this paper, we consider the time that the first node takes to run out of energy as the whole network's lifetime.

In fact, bottleneck nodes will consume energy no matter which node we choose as the preferred parent. In order to ensure a more efficient evaluation of whether the bottleneck node is energy-balancing, it is necessary to evaluate the degree of dispersion of the remaining energy consumption of these nodes. Based on the relevant mathematical knowledge, we use range, average deviation, and standard deviation to assess the degree of data dispersion. Nevertheless, through many tests, we found that these metrics could not embody the distribution of the remaining energy consumption of the bottleneck nodes.

To estimate the distribution of the energy consumption of the bottleneck nodes precisely, this paper provides the measurement of the energy distribution equilibrium degree based on Pareto's evaluation model [27].

The definition of the energy of dispersion: Assume the rational number set represents sensor energy, $a_i \geq 0$, and the ED (energy of dispersion) criteria can be:

$$ED = \sum_{i=1}^{n} \varpi_i \cdot {a_i}^2 \tag{5}$$

Here, $\varpi_i = \frac{1/(a_i - E(A))^2}{1/D(A)}$, $D(A) = \|A - E(A)\|_2/n$, $E(A) = \|A\|_1/n$, $\|\cdot\|_x$ represents the norm of x.

To validate the effectiveness of ED, we compare some typical measurements for evaluating the dispersion degree, including the range, average deviation and standard deviation.

Firstly, we demonstrate numerical evaluations to compare their differences. A total of 200 numbers from 1 to 100 were randomly selected, and these measure values were calculated individually. The test was run 1000 times and recorded the maximum value, the minimum value and the average value. The result can be seen in Table 2 and Figure 4.

Table 2. Comparison of several discrete degree metrics.

Measurement	Maximum	Minimum	Average
Range	98.6920	96.3815	97.5278
Average Deviation	40.4198	9.3249	26.5953
Standard Deviation	28.7187	9.3249	28.5178
ED	1.4985×10^4	3.9625×10^3	8.6994×10^4

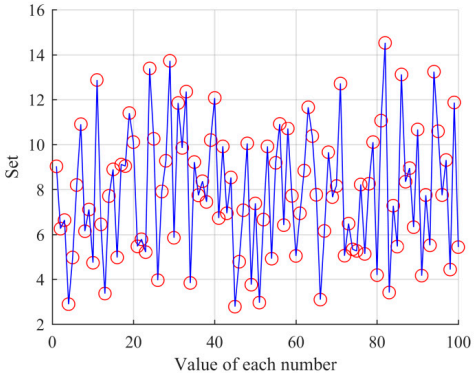

Figure 4. Distribution of values.

The above table shows that the ED can express the data dispersion degree quite well and reflects the fluctuation degree more clearly.

Furthermore, we compared these measurement standards based on the RPL protocol. There were 100 nodes in the tests and each node has the same initial energy. We converted the energy into 100 units for simplification.

According to the RPL protocol, the preferred parent node needs to be selected first. For each possible parent node (that is, where the rank of the neighbor is smaller than its own), a node will take the following steps:

Step 1: Calculate the measurement set of the bottleneck nodes (range (R), average deviation (AD), standard deviation (SD) and energy of dispersion (ED)) through the parent nodes broadcast. This will send all the traffic to that parent node and save the minimum value (Line 4);

Step 2: Calculate the own lifetime and make sure it will not become another new bottleneck node if chosen as the father node (Line 5);

Step 3: Remove the traffic that arrives at that parent node and test other nodes. Make sure all the parent nodes have been tested before making the decision (Line 10);

Step 4: Select the preferred parent nodes. The node maximizes the lifetime of the bottleneck with the minimum lifetime (Line 6, 7, 8).

An outline is listed in Algorithm 1.

Algorithm 1 Evaluation of different metrics based on RPL protocol

Input: The sender node a; the candidate parent set of a, *parent*(a); the bottleneck set, *C*; and its number *N*
Output: the preferred parent of a
Initialization *min_MD* = 100,000;

1. **for** $b \in parent(a)$
2. $\alpha_{a,b} = 1$

// test every parent of a

3. Obtain all bottleneck nodes' residue energy, $E_{rse}(c_i)$, $c_i \in C$, $i = 1 \cdots N$
4. Compute $E(C) = \sum_{i=1}^{N} E_{rse}(c_i)/N$, $D(C) = \sum_{i=1}^{N} (E_{rse}(c_i) - E(C))^2$, $\varpi_i = \frac{1/(a_i - E(C))^2}{1/D(C)}$, then obtain the range $R = \max\{E_{rse}(C)\} - \min\{E_{rse}(C)\}$, the average deviation $AD = \sum_{i=1}^{N} |E_{rse}(c_i) - E(C)|/N$, the standard deviation $SD = \sqrt{D(C)}$, and our metric $ED = \sum_{i=1}^{n} \varpi_i \cdot (E_{rse}(c_i))^2$)

// compute the range, average deviation, standard deviation, and *our metric of the bottleneck set C*

5. **If** *min_MD* > R (or AD, SD, ED) **then**
6. *min_MD* = R (or AD, SD, ED);
7. preferred_parent = b;
8. *// select the current optimal parent*
9. **end**
10. $\alpha_{A,B} = 0$

// continue to test the next parent

11. **end**

The above algorithm is used to choose preferred parent nodes based on single-path circumstances. We should set min_MD to be as large as possible to guarantee that the value will not be smaller than the ED value under real circumstances.

When the first node has used up its energy, the experiment stops and records the bottleneck nodes' energy. We achieved the average value through 1000 cycles; the results are in Table 3.

Table 3. The comparison results for several dispersion degrees.

Measurement	Maximum	Minimum	Average
Range	71.0600	69.9184	71.0588
Average Deviation	27.7367	6.1617	27.7151
Standard Deviation	28.8025	6.1617	28.8007
ED	1.0438×10^5	574.5076	678.3107

The evaluation verifies that the ED can evaluate well the degree of imbalance of a network node's energy consumption.

3.1.3. Multi-Path Energy Consumption

If we use $\alpha_{x,y}$ as the traffic ratio of node x sent to its parent node y, then the bottleneck node's traffic is the traffic ratio that the node sends to three parent nodes individually, multiplying the traffic ratio that they retransmit to bottleneck nodes, and then adding them. More generally, the expression should be:

$$\alpha_{a,c} = \sum_{b \in Parents(a)} \alpha_{a,b} \times \alpha_{b,c} \tag{6}$$

We analyze the bottleneck nodes' energy consumption based on multi-path distribution strategy in this section. As we know, nodes will send packets to several parent nodes through links of different quality, namely, nodes will transmit traffic through several paths, and hence, the energy that will be consumed will depend on the traffic sent to the each parent node. First, we elaborate how to calculate the bottleneck nodes' remaining energy consumption under a multi-path environment.

In Figure 5, node A is the sending node, its parent nodes are B1, B2 and B3, and the bottleneck nodes are C1 and C2. For node C1, it receives 100% of the traffic of B1 and 20% of B2. The real traffic of C1 is 30% + 50% × 20%. Similarly, the real traffic of C2 is 20% + 50% × 80%.

$$Tat(a,c) = \sum_{b \in Parents(a)} \alpha_{a,b} \times \frac{Tat(a,b)}{PRR(a,b)} \times \alpha_{b,c} \times \frac{Tat(b,c)}{PRR(b,c)} \tag{7}$$

Thus, in terms of bottleneck node C1, Equations (1) and (3) will be:

$$E_{rce}(c_1) = Tat(a,c_1) \times \frac{E_r}{D_r} \tag{8}$$

$$E_{tat}(c_1) = \sum_{d \in Parents(c_1)} \frac{\alpha_{c_1 \cdot d} \times Tat(c_1,d)}{PRR(c_1,d)} \times \frac{E_t}{D_d} \tag{9}$$

In this case, the energy consumption of bottleneck node C1 is:

$$\begin{aligned} E_{rce}(c_1) + E_{cmp}(c_1) + E_{tat}(c_1) = \quad & Tat(a,c_1) \times \tfrac{E_r}{D_r} + N_{task}(Buffer\_Use(c_1)) \times E_c + \\ & \textstyle\sum_{d \in Parents(c_1)} \frac{\alpha_{c_1,d} \times Tat(c_1,d)}{PRR(c_1,d)} \times \frac{E_t}{D_d} \end{aligned} \tag{10}$$

The parameter in the above equation can be obtained before transmission, except for the distribution traffic ratio, thus node A can assess the impact of the survival time of the bottleneck node.

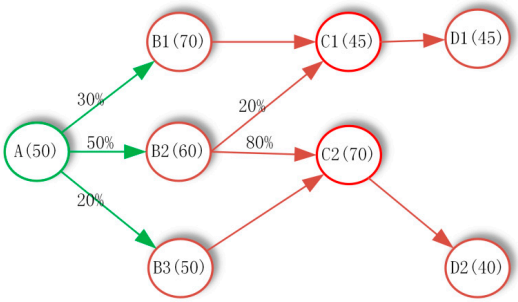

Figure 5. Multi-path energy consumption.

Each node can calculate the transmission ratio of arriving at bottleneck nodes in a recursion manner. Thus the real traffic that the bottleneck nodes obtain is multi-path traffic.

3.2. Multi-Path Traffic Distributions

According to RPL, nodes will separate traffic on every path available to guarantee that the lifetime of each bottleneck node is balanced. To maximize the network lifetime, we considered using a plain linear program to assign weights to each parent node to obtain the optimal solution. However, this is apparently not suitable for sensor nodes that are limited in both storage and calculation resources.

This paper proposes a quick algorithm based on the metric mentioned above. Firstly, we convert metric ED to an objective function (ED = f($a_{x1,c1}$, $a_{x2,c2}$, ...)) about $a_{x,c}$, depending on correlative mathematical knowledge, where c_i belongs to the bottleneck node set, and x_i belongs to the descendants of c_i. The function is of the two order continuous differentiable function, which can be solved by linear programming. Then, we combine Newton's as well as the steepest descent method to propose one modified algorithm. This method was able to find one of the best group weights to send traffic by an iteration test. At the same time, it can realize convergence as soon as possible and has quite good stability and requires little calculation. Algorithm 2 provides its formal description: the nodes decide whether it receives a proper answer (Line1), if not, it is judged as to whether or not an accurate search can be performed (Line 2). If it can, the direction would be $-\frac{\nabla f(x_k)}{\nabla^2 f(x_k)}$ (Line 3). If it can not, it uses a negative gradient search direction $-\nabla f(x_k)$ (Line 5). Then, the step length is adjusted (Line 7), acting on iteration (Line 8), until we get the optimum solution.

Algorithm 2 Energy balance based on a multi-path

Input: Object function $f(\alpha_{a,b_1}, \alpha_{b,c_2}, \cdots)$
Output: $\langle \alpha_{b,c_1}, \alpha_{b,c_2}, \cdots \rangle$ for energy balance
Initialization $\delta > 0$, $\alpha_{b,c}$ = random [0, 1] and $\sum_{c \in bnk(b)} \alpha_{b,c} = 1$, $x_0 = \langle \alpha_{b,c_1}, \alpha_{b,c_2}, \cdots \rangle$, $k = 0$

1. **for**$||\nabla f(x_k)|| > \delta$
2. **if** $\frac{\nabla f(x_k)^T \cdot \nabla f(x_k)}{\nabla^2 f(x_k)} > 0$ **then**
3. $d_k = -\frac{\nabla f(x_k)}{\nabla^2 f(x_k)}$
4. **else**
5. $d_k = -\nabla f(x_k)$
6. **end**
7. $\lambda_k = -\frac{-\nabla f(x_k)^T \cdot d_k}{d_k{}^T \cdot \nabla^2 f(x_k) \cdot d_k{}^T}$;
8. $x_{k+1} = x_k + \lambda_k \cdot d_k$
9. $k = k + 1$;
10. **end**
11. **Return** x_k

The advantage of the algorithm is that the iterative step-size adjustment uses the precision searching method to minimize the quadratic function. The convergence rate is linear and can be guaranteed every time. When the maximum eigenvalue and the minimum eigenvalue of the Hessian matrix are close to each other, the descending velocity is the fastest. Particularly, when they are equal to each other, we obtain the optimum solution through only one iteration. Furthermore, the number of bottleneck nodes changes with the network. When the number of bottlenecks increases, the energy consumption of x increases. At this time, we can use the quasi-Newton's method to reduce the amount of calculation for the Hessian matrix, namely,

$$\nabla^2 f(x_{k+1}) = \nabla^2 f(x_k) + \frac{s_k (s_k)^T}{(s_k)^T y_k} \left[1 + \frac{(y_k)^T \cdot \nabla^2 f(x_k) \cdot y_k}{(s_k)^T y_k} \right] - \frac{1}{(s_k)^T y_k} \left[s_k (y_k)^T + (s_k)^T y_k \right] \cdot \nabla^2 f(x_k) \quad (11)$$

where $s_k = x_{k+1} - x_k$, $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$.

3.3. End-to-End Delay Optimization

This section studies how a node's cache impacts end-to-end delay. We abstract the transmission procedure as the Markov procedure and analyze the packet loss depend on bottleneck nodes' remaining cache. Subsequently, we calculate the dispersion degree of the remaining cache size of the bottleneck nodes. Finally, we provide the multi-path data distribution algorithm depending on the remaining cache size of the bottleneck nodes.

3.3.1. Data Forwarding Model and Node Transfer Latency

The limited capacity of the cache is a vital factor that influences network performance. The efficiency of the cache directly affects the reliability and stability of the entire network, and also indirectly affects its survival time. The unreasonable routing protocol always leads to data gathering together at some nodes, which makes their cache overflow. More seriously, they can only receive data from descendant nodes, and can not be forwarded further. This "invalid operation" node greatly increases the end-to-end transmission time. On the other hand, if some nodes' cache utilization rates are too high, that is, there are too many data frames in the node, and then its CPU (Central Processing Unit) will spend too much energy on receiving, handling and sending data. If these conditions continue, a certain number of nodes will soon fail, resulting in network vacancies, which seriously undermine the network connectivity, extending the end-to-end delay.

In Figure 6, S1 to S4 represent the data sources, and R1 to R3 represent the relay nodes. If all the data sources select R2 to do data transmission, then that node will gradually become a bottleneck node. Due to the restrictions of the cache, it must lose some data if the cache utilization rate achieves a certain level.

This paper calculates the nodes' delay through cache utilization. Let the size of the nodes' cache be S. Assume the data arriving obeys the Markov procedure and that the transmission of relay nodes also obeys the Markov procedure, then we can abstract the transmission model to a M/M/1/S/FCFS model. If the data packet arriving at the nodes can not be placed into the cache pool, these packets will be lost. According to the queuing theory, the nodes' cache can have the desired value, such as:

$$L = \begin{cases} S/2 & \delta = 1 \\ \frac{\delta\left[1-(S+1)\delta^S+S\delta^{+1}\right]}{(1-\delta^{S+1})(1-\delta)} & \delta \neq 1 \end{cases} \tag{12}$$

Here, $\delta = S_r / D_p \cdot T_p \cdot N_{task}$.

The remaining cache R is $R = S - L$. The remaining cache has a significant impact on the loss of packets. A node's packet loss ratio is:

$$p = \begin{cases} 1/(R+1) & \delta = 1 \\ \frac{\delta^R(1-\delta)}{1-\delta^{R+1}} & \delta \neq 1 \end{cases} \tag{13}$$

Combining L with P, the resulting data packet's average waiting time is:

$$T = \frac{L}{S_r(1-p)} - \frac{1}{D_p \cdot T_p \cdot N_{task}} \tag{14}$$

The waiting time of a data packet is determined by L, P, S_r, D_p, T_p, N_{task}. L and P can be solved by the last four parameters. The last three parameters are determined by sensors, that is, they are constants. The average waiting time is mainly caused by the amount of receiving data. This means that a reasonable allocation of traffic can reduce the end-to-end transmission time.

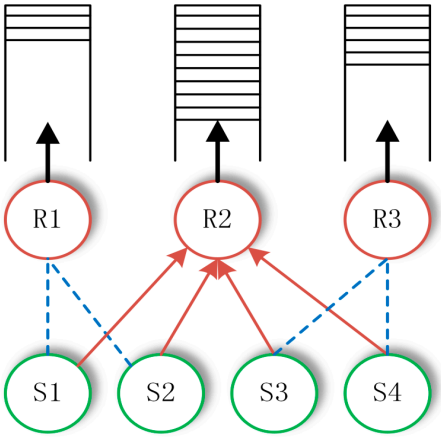

Figure 6. Uneven cache usage.

3.3.2. Waiting Time for Multi-Path Transmission Nodes

The delay of end-to-end transmission is mainly based on the waiting time of a data packet in the bottleneck nodes. As Figure 5 shows, the bottleneck nodes' traffic is the ratio of traffic that node A sent to its three parent nodes individually, multiplied by the ratio of traffic that is transmitted to the bottleneck, and then added. Hence, the number of data received by the bottleneck node in the multi-path case is:

$$S_r(a,c) = \alpha_{a,c} \cdot Tat(a,c) \tag{15}$$

Thus, its waiting time D is:

$$D = T_{a,c} = \frac{L}{S_r(a,c)(1-p)} - \frac{1}{D_p T_p N_{task}} \tag{16}$$

3.3.3. Multi-Path Traffic Distribution

According to the above model, we propose a multi-path traffic distribution algorithm based on RPL to reduce the waiting time. This algorithm makes use of greedy thoughts. A heuristic scheme is used to determine the weight of each associated parent node. Specifically, the divided nodes distribute the traffic into N parts. Each part assigns to its parent node which can minimize the maximum delay among all of the bottleneck nodes.

Algorithm 3 provides the formal description, the main idea is to test all the parent nodes by greedy iteration, and find out the approximate optimal flow distribution scheme. Take Figure 3 as an example. Node A first allocates traffic evenly for each parent node, and then tests the smallest of the maximum waiting times in the bottleneck node by gradually adjusting the weight of the parent node. If the maximum waiting time can be reduced, it saves the current node and reinitializes traffic distribution so that other parent node can be tested. Finally, it assigns a step length to the most appropriate parent node and restarts the iteration, and finds a reasonable flow distribution.

The selection of the step size determines the optimal result of the flow distribution and the energy consumption of the node. Using a small step length can be able to obtain less waiting time for the bottleneck nodes, while strengthening the complexity of the calculation. Conversely, the traffic distribution scheme is not reasonable enough. That is, we should set a proper step length with regard to the real situation and request.

Algorithm 3 End-to-end delay based on multi-path

Input: The sender node A and its transmission quantity; the candidate parent set of A, parent(A) and its number; the bottleneck set, C, and its number N; $\Delta\alpha$
Output: all $\alpha_{a,b}$

1. **for** i = 1 to $1/\Delta\alpha$
2. min_time = 10,000;
3. All $\alpha_{a,b} = Tat(a)/Num(parent(a))$;
4. **for** $b \in parent(a)$
5. $\alpha_{a,b} = \alpha_{a,b} - \Delta\alpha$
   // *test every step*
6. obtain all bottleneck nodes' waiting time $T(c_i)$, $c_i \in C$, $i = 1 \cdots N$, then
   $max\_time = \max\{T(c_i)\}$
7. **if** $max\_time < min\_time$ **then**
8. $max\_time = min\_time$;
9. min = b;
10. **end**
11. $\alpha_{a,b} = \alpha_{a,b} + \Delta\alpha$
12. **end**
13. $\alpha_{\min} = \alpha_{\min} - \Delta\alpha$
14. **end**

3.3.4. Adaptive Traffic Assignment Algorithm

The above section considers the energy balance and the end-to-end delay in a multi-path situation, separately. The focus may be different in some practical applications. For instance, we should think more about the end-to-end delay when we facing the real-time circumstances, while for normal surveillance data, we pay more attention to maximizing the network lifetime. Thus, this paper proposes adaptive multipath traffic loading based on RPL. This algorithm chooses different objective functions according to the different requirements. We set out the details in Algorithm 4. We use Algorithm 2 (Line 2) if we consider the energy balance. If the energy balance is achieved at a certain threshold, then we adjust the ratio of data distribution based on bottleneck node's waiting time to shorten the end-to-end delay (Line 4). If we consider the end-to-end delay first, we use Algorithm 3 (Line 6); when the delay is smaller than the determined threshold, then we change to Algorithm 2.

Algorithm 4 Adaptive Multipath Traffic Loading

Input: Γ_{ed} and Γ_{de} // Γ_{ed} is the threshold for energy dispersion of the network, Γ_{de} is the threshold utilization rate for delay of bottleneck.
Output: $\langle a_{b,c_1}, a_{b,c_2}, \cdots \rangle$

1. Case energy-first:
2. Call Algorithm 2
3. If ED < Γ_{ed} then
4. Call Algorithm 3
5. Case delay-first:
6. Call Algorithm 3
7. **If** D < Γ_{de} **then**
8. **Call** Algorithm 2
9. **end**

4. Analysis of the Performance

This section evaluates the performance of our algorithm. It mainly proves the network performance (load distribution, network lifetime, energy consumption, end-to-end delay, and stability of the route) using the former algorithm under different network sizes or with different caches in nodes.

4.1. Evaluation Environment

The experimental parameters are set as shown in Table 4.

Table 4. Parameter setting.

Parameter	Value
Number of nodes	At most 89
Area	100 × 100 m
communication radius	10–20 m
Flow patterns and rates	CBR, 5 pkt/min
The size of packet	126 bytes
Duration time	1 h
Number of bottleneck	5
Minimum step size of RPL	MinHopRankIncrease = 128
Trickle	Imin = 2^7 ms, Imax = 16 ms, k = 10
MAC	IEEE 802.15.4
Energy consumption model	According to CC2530 data sheet

4.2. Analysis of the Network Performance

4.2.1. Network Load Distribution with Different Buffer Size

First, we compare the distributions of the network load with different buffer sizes in nodes. Data packages are sent every five minutes and the sink node is at the center of the network. The residual energy is shown in the following figures. The node cache size of Figure 7a is 30 bytes. The node cache size of Figure 8a is 50 bytes. From the figures we can see that sensor nodes that are closer to the sink always have a heavier workload than edge nodes, which is consistent with the characteristics of sensor network convergence transmission, because all traffic flows to the sink node. The comparison between Figures 7a and 8a also found that there is an obvious peak in the central portion when the node has a smaller buffer size. This is because the possibility of retransmission rises due to the loss of data. The smaller the cache is, the easier for it to be full for the nodes closed to the center, which causes the loss of data. Figure 7a,c and 8a,c compare the energy consumption between Algorithms 2 and 4. From the figures we can see that the network lifetime will be longer if we use the energy balance to make the distribution of the energy more uniform. There is no need to consider the energy consumption of the sink node, since the sink node usually uses the mains-powered system instead of the battery-powered system. We can also find from the figures that Algorithm 2 has the best result. That is because it first considered the energy balance without end-to-end delay as the main factor. At the same time, the curve of this test is smooth. It shows that the load of nodes is almost the same if they are of the same distance to the sink node. However, RPL cannot ensure this. As a result, our algorithm can efficiently balance the load of the packets which are sent by the network. It can better ensure the load balance if there are less nodes. In the actual situation, as time goes, there will be some bottleneck. Their processing speed will be slowing down obviously, resulting in the cache of the surrounding nodes to drop. Using the algorithm we discussed in this essay will effectively remit the tension level of bottleneck nodes and the surrounding nodes.

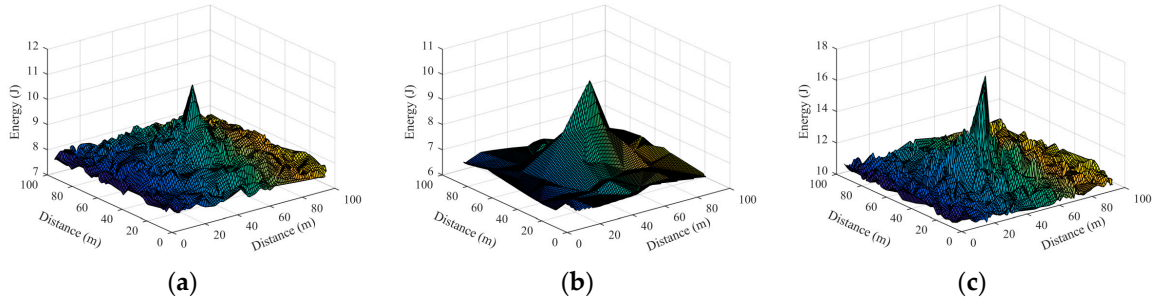

Figure 7. Residual energy distribution (cache size 30). (**a**) For original RPL; (**b**) For Algorithm 2; (**c**) For Algorithm 4.

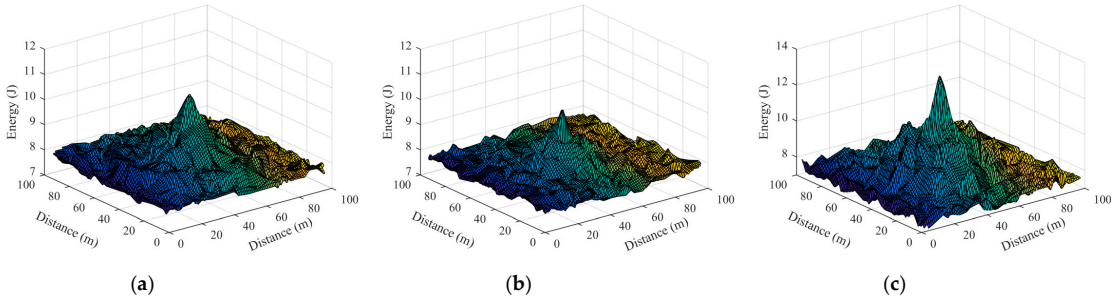

Figure 8. Residual energy distribution (cache size 50). (**a**) For original RPL; (**b**) For Algorithm 2; (**c**) For Algorithm 4.

4.2.2. Network-Lifetime

We validated the longest lifetime for Algorithms 2 and 4 using different network sizes. Figure 9 shows that our algorithm can achieve a longer lifetime. This advantage is more obvious especially when the network size is not large. Algorithm 2 can provide the longest lifetime, and Algorithm 4 is a bit weaker than Algorithm 2. This is mainly because when the energy balance reaches a certain threshold, the node will choose the multi-path forwarding according to the cache utilization ratio, and a new energy imbalance will occur. However, it will certainly repeat the energy balance algorithm to achieve energy balance. However, it should be noted that with the increase in the number of network nodes, the network survival time suddenly dropped. The nodes around the sink node have to use a lot of energy to transmit the data. By this moment, none of these three algorithms can greatly reduce energy consumption.

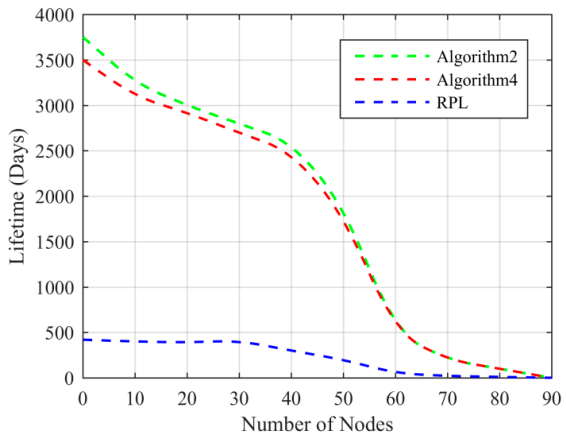

Figure 9. Network lifetime in different size.

4.2.3. Analysis of the End-to-End Delay

Subsequently, we compared the delay between the nodes based on Algorithm 3, Algorithm 4 and the original RPL in Figure 10. According to the results, Algorithm 3 needs the least time because it only focuses the transmission time between the nodes and use the best route to do the timely data forwarding. It also extremely reduces the possibility of the loss of data and the low probability of data retransmission. Although Algorithm 4 is not as effective as Algorithm 3, its delay is still smaller than the original RPL. It firstly considers the energy balance and ensures the network lifetime for as long as possible. On the basis of this, it considers the cache utilization. However, the original RPL always carries out packet forwarding by choosing the preferred parent node. The choice of the parent node is determined by the objective function, which is determined by certain metrics. As long as the topology does not change, the preferred parent node will not change. There is no guarantee of the end-to-end delay.

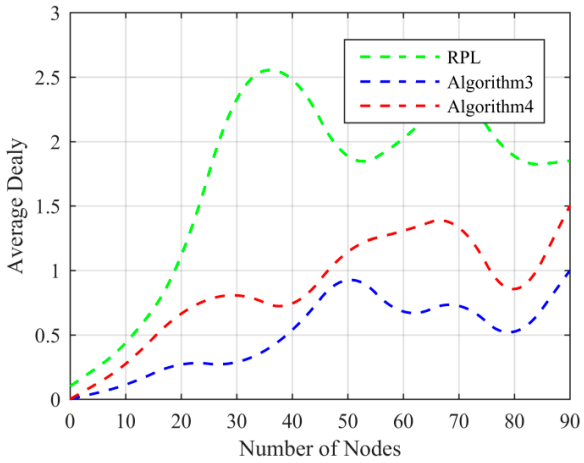

Figure 10. Network end-to-end delay in different size.

From the figure we can see that there is irregular fluctuation of delay under different network sizes. It responds to the position of the nodes.

4.2.4. The Stability of the Route

The stabilities of Algorithms 2–4 are shown in Figure 11. From the figure, the original RPL has the least stability. This is because the original will frequently change the preferred parent node, which results in the instability of the network and excessive energy consumption. In fact, the change of the preferred parent node will cause a reset of the trickle timer, which will make the transfer of control messages more frequent. The more we transfer the control messages, the more energy it will consume. Moreover, the frequent changes will make the network unstable and will worsen the network topology. The above algorithms, except the original RPL, make all parent nodes participate in traffic forwarding without changing the preferred parent node. The preferred parent node will only change if it fails, and then the trigger timer will be reset. This will ensure network stability.

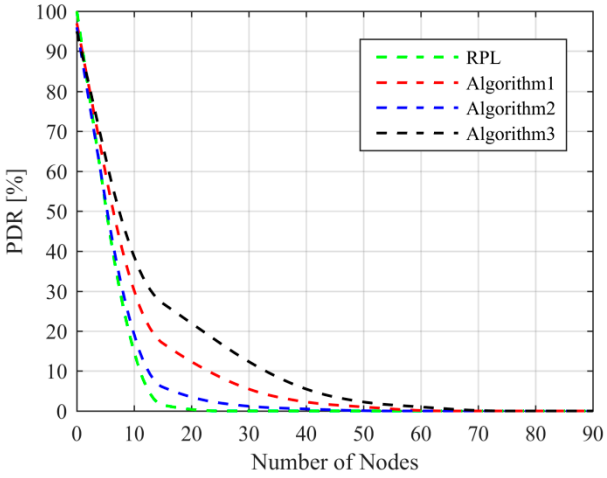

Figure 11. PDR (Packet Deliver Rate) in different network size.

5. Conclusions

In this paper, we proposed an adaptive multipath traffic loading based on RPL to meet the needs of energy consumption and end-to-end delay. First, we designed an energy-balanced routing algorithm to improve the network lifetime. By establishing a practical model and proposing a standard of energy dispersion degree measurement, it can effectively determine the degree of energy balance of nodes. Based on this metric, we present a fast algorithm to obtain the optimal distribution of the data. The algorithm maintains the stability of the parent set, thus improving the stability of the network.

Furthermore, in order to decrease end-to-end delay, we converted the data transmission model to a M/M/1/S/FCFS model based on cache utilization, calculated the average waiting time of data, and then obtained the waiting time of the bottleneck node in the multi-path. Using the calculated time value, we proposed a traffic allocation algorithm based on greedy thought, which can optimize the end-to-end delay.

Finally, taking into account the application needs of different scenarios, we integrated the two above algorithms. If the network prioritizes the survival time, priority is given to the energy balance, and then the end-to-end delay. If end-to-end delay is the main factor of the network, we consider that the cache utilization decreases the loss of packets. The results indicate that the three schemes all improve the reliability of routing, increase the network lifetime, decrease the end-to-end delay and decrease the time for DAG resetting.

It can be seen from the experiment that the nodes around the sink still consume more energy due to their bearing more traffic. They are still the bottlenecks of the entire network. In future work, the nodes around the sink will be optimized to maximize their survival time. Such measures include deploying more nodes around the sink, increasing their lifetime by turning off the radio, increasing the transmission distance of a single hop, and even using mobile sink nodes to make the energy consumption of the entire network balance, extending the survival time of the network.

Acknowledgments: This work was supported in part by Natural Science Research and Development Projects of Jiangsu Province (No. BY2016066-05), the National Science Foundation of China (No. 61379064, 61572260, 61572261, 61772448), the National Science Foundation of Jiangsu Province of China (No. BK20140462).

Author Contributions: Licai Zhu designed all the algorithms and wrote the paper. Ruchuan Wang made a careful revision of the article and proposed amendments. Hao Yang performed the experiments and analyzed the data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, K.F. Smart home technology for telemedicine and emergency management. *J. Ambient Intell. Humaniz. Comput.* **2012**, *4*, 535–546. [CrossRef]
2. Sanislav, T.; Mois, G.; Folea, S.; Miclea, L. A cloud-based Cyber-Physical System for environmental monitoring. In Proceedings of the 3rd Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 15–19 June 2014; pp. 6–9.
3. Wang, S.; Xing, J.; Li, J.; Yang, Q. A decentralized flat control system for intelligent building. In Proceedings of the 27th Control and Decision Conference (CCDC), Qingdao, China, 23–25 May 2015; pp. 2622–2627.
4. Thubert, P.; Winter, T.; Brandt, A.; Hui, J.; Kelsey, R.; Levis, P. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. March 2012. Available online: https://www.rfc-editor.org/info/rfc6550 (accessed on 6 May 2017).
5. Andrea, K.; Simon, R. Design and Evaluation of an RPL-based Multi-Sink Routing Protocol for Low-Power and Lossy Networks. In Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Cancun, Mexico, 2–6 November 2015; pp. 141–150.
6. De Couto, D.S.J.; Aguayo, D.; Bicket, J.; Morris, R. *A High-Throughput Path Metric for Multi-Hop Wireless Routing*; Springer Wireless Networks: London, UK, 2005; pp. 419–434.
7. Conti, M.; Gregori, E.; Maselli, G. Reliable and efficient forwarding in ad hoc networks. *Ad Hoc Netw.* **2006**, *4*, 389–415. [CrossRef]
8. Radi, M.; Dezfouli, B.; Bakar, K.A.; Lee, M. Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges. *Sensors* **2012**, *12*, 650–685. [CrossRef] [PubMed]
9. Hong, K.S.; Choi, L. DAG-based multipath routing for mobile sensor networks. In Proceedings of the ICT Convergence (ICTC), Seoul, Korea, 28–30 September 2011; pp. 261–266.
10. Nurmio, J.; Nigussie, E.; Poellabauer, C. Equalizing energy distribution in sensor nodes through optimization of RPL. In Proceedings of the 15th IEEE International Conference on Computer and Information Technology, CIT 2015, Liverpool, UK, 26–28 October 2015; In Proceedings of the 14th IEEE International Conference on Ubiquitous Computing and Communications, IUCC 2015, Liverpool, UK, 26–28 October 2015; In Proceedings of the 13th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2015, Liverpool, UK, 26–28 October 2015; In Proceedings of the 13th IEEE International Conference on Pervasive Intelligence and Computing, PICom 2015, Liverpool, UK, 26–28 October 2015. pp. 83–91.
11. Chen, Y.; Nasser, N. Energy-balancing multipath routing protocol for wireless sensor networks. In Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, Waterloo, ON, Canada, 7–9 August 2006; pp. 21–24.
12. Yahya, B.; Ben-Othman, J. REER: Robust and Energy Efficient Multipath Routing Protocol for Wireless Sensor Networks. In Proceedings of the Global Telecommunications Conference, Honolulu, HI, USA, 30 November–4 December 2009; pp. 1–7.
13. Liu, X.; Guo, J.; Bhatti, G.; Orlik, P. Load balanced routing for low power and lossy networks. In Proceedings of the Wireless Communications and Networking Conference (WCNC), Shanghai, China, 7–10 April 2013; pp. 2238–2243.
14. Moghadam, M.N.; Taheri, H. High throughput load balanced multipath routing in homogeneous wireless sensor networks. In Proceedings of the 22nd Iranian Electrical Engineering (ICEE), Tehran, Iran, 20–22 May 2014; pp. 1516–1521.
15. Pavković, B.; Theoleyre, F.; Duda, A. Multipath opportunistic RPL routing over IEEE 802.15.4. In Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Miami, FL, USA, 31 October–4 November 2011; pp. 179–186.
16. Grubman, T.; Şekercioğlu, Y.A.; Moore, N. Opportunistic Routing in Low Duty-Cycle Wireless Sensor Networks. *ACM Trans. Sens. Netw.* **2014**, *10*, 1–39.
17. Kacimi, R.; Dhaou, R.; Beylot, A.L. Load balancing techniques for lifetime maximizing in wireless sensor networks. *Ad Hoc Netw.* **2013**, *11*, 2172–2186. [CrossRef]
18. Iova, O.; Theoleyre, F.; Noel, T. Using multiparent routing in RPL to increase the stability and the lifetime of the network. *Ad Hoc Netw.* **2015**, *29*, 45–62. [CrossRef]
19. Chang, J.H.; Tassiulas, L. Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Trans. Netw.* **2004**, *12*, 609–619. [CrossRef]

20. Systems, C.; Kim, M.; Pister, K.; Dejean, N.; Barthel, D. Routing Metrics Used for Path Calculation in Low-Power and Lossy Networks. March 2012. Available online: https://www.rfc-editor.org/info/rfc6551 (accessed on 20 April 2017).
21. Liu, H.; Zhang, Z.L.; Srivastava, J.; Firoiu, V. PWave: A Multi-source Multi-sink Anycast Routing Framework for Wireless Sensor Networks. In Proceedings of the 6th International IFIP-TC6 Networking Conference, Atlanta, GA, USA, 14 May 2007; pp. 179–190.
22. Chang, L.H.; Lee, T.H.; Chen, S.J.; Liao, C.Y. Energy-Efficient Oriented Routing Algorithm in Wireless Sensor Networks. In Proceedings of the Systems, Man, and Cybernetics (SMC), Manchester, UK, 13–16 October 2013; pp. 3813–3818.
23. Le, Q.; Ngo-Quynh, T.; Magedanz, T. RPL-based multipath routing protocols for Internet of Things on wireless sensor networks. In Proceedings of the Advanced Technologies for Communications (ATC), Hanoi, Vietnam, 15–17 October 2014; pp. 424–429.
24. Hossain, A.K.M.; Sreenan, C.J.; Alberola, R.D.P. Neighbour-disjoint multipath for low-power and lossy networks. *ACM Trans. Sens. Netw.* **2016**, *12*, 23. [CrossRef]
25. Barcelo, M.; Correa, A.; Vicario, J.L.; Morell, A. Cooperative interaction among multiple RPL instances in wireless sensor networks. *Comput. Commun.* **2016**, *81*, 61–67. [CrossRef]
26. Gonizzi, P.; Monica, R.; Ferrari, G. Design and evaluation of a delay-efficient RPL routing metric. In Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013; pp. 1573–1577.
27. Mohamed, B.; Mohamed, F. QoS Routing RPL for Low Power and Lossy Networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 1–10. [CrossRef]
28. Gaddour, O.; Koubâa, A.; Abid, M. Quality-of-service aware routing for static and mobile IPv6-based low-power and lossy sensor networks using RPL. *Ad Hoc Netw.* **2015**, *33*, 233–256. [CrossRef]
29. Michel, M.; Duquennoy, S.; Quoitin, B. Load-Balanced Data Collection through Opportunistic Routing. In Proceedings of the Distributed Computing in Sensor Systems (DCOSS), Fortaleza, Brazil, 10–12 June 2015; pp. 62–70.
30. Hao, X.C.; Wang, M.Q.; Hou, S.; Gong, Q.Q.; Liu, B. Distributed Topology Control and Channel Allocation Algorithm for Energy Efficiency in Wireless Sensor Network: From a Game Perspective. EBSCO. *Wirel. Pers. Commun.* **2015**, *80*, 1557–1577. [CrossRef]