

Research Article

Location-Based Data Aggregation in 6LoWPAN

Jinyu Hu,¹ Juan Luo,¹ Yuxi Zhang,² Panwu Wang,¹ and Yu Liu¹

¹College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan 410082, China

²Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

Correspondence should be addressed to Juan Luo; juanluo@hnu.edu.cn

Received 20 October 2014; Revised 9 March 2015; Accepted 20 March 2015

Academic Editor: Xiaohong Jiang

Copyright © 2015 Jinyu Hu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Location-based information has recently been exploited to assist the aggregated process of data, thereby reducing the spatial redundancy efficiently. The constraints nature in 6LoWPAN becomes one of the major concerns in data aggregation methods. However, traditional CSMA/CA in MAC layer may cause significant transmission and control overhead as well as delay on listening and competing for channels. It is a low efficient way to transfer IPv6 packet due to the big packet header. To overcome these shortages, in this paper, we propose LDAA, a location-based novel data aggregation model that aggregates data from the network layer according to the MAC layer queuing delay. When the queuing delay becomes larger, more packets will be dynamically aggregated into one packet to increase the proportion of application data. Otherwise, the amount of packets involved in aggregation will decrease to improve channels utilization. Simulation results show that our approach could provide better real-time guarantees and reduce data spatial redundancy and energy consumption efficiently.

1. Introduction

Motivation. The Internet of Things (IoT) [1] is a novel application that has gained extensive attention in modern wireless telecommunication scenarios. The main idea of IoT is the pervasive presence around us of a variety of embedded devices, such as sensors and Radio-Frequency Identification (RFID) tags. Through unique addressing schemes, these devices are able to interact with each other and cooperate with their neighbors to reach common goals [2]. However, the limited resources of embedded devices, such as energy, computing ability, and storage capacity, have greatly restricted the development of IoT applications. Meanwhile, the communication mechanism between heterogeneous embedded devices and Internet servers is still a challenge for IoT.

As a solution for the above problems, 6LoWPAN [3] (IPv6 over Low Power Wireless Personal Area Network) firstly introduces IPv6 to the wireless personal area network. This technology combines the IEEE 802.15.4 MAC layer with the IPv6 network layer and effectively solves the communication problem between heterogeneous embedded devices and Internet servers. In 2008, with the further research on low

power IPv6 network protocol, ISA formulated the SP100.11A standard for 6LoWPAN. Also, IP500 Union has begun to promote the development of 6LoWPAN technology. 6LoWPAN networks, at present, have emerged as an essential part of people's life in various fields, such as automotive industry, health care, environmental monitoring, and agriculture surveillance. As a special kind of wireless sensor network, 6LoWPAN networks have the following characteristics.

- (1) Massive nodes are deployed to detect and evaluate the events happened in some certain situations.
- (2) Very constrained power, bandwidth, and memory of nodes may seriously affect the network performance.
- (3) Upper-layer protocol data unit may get encapsulated by using IPv6 header.
- (4) Specific applications need to frequently collect and transmit data in their monitored area.
- (5) Different applications require different QoS, which make packets have different priorities.

However, these characteristics of 6LoWPAN networks can easily lead to data redundancy. As a consequence, these

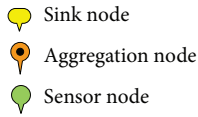


FIGURE 1: Agriculture surveillance system.

redundant data may quickly exhaust the energy of sensor nodes. For example, agriculture surveillance system is a typical application of IoT, as shown in Figure 1. In a tea plantation scenario, hundreds of specialized sensors are randomly deployed in the monitored area, collect the targets' temperature, humidity, and sunlight data constantly, and then transmit these sensing data to the sink node periodically. In order to obtain reliable and accurate information about tea plants in real time, data collection must be generated at a short interval. And the more simultaneous the data transmit, the easier the congestion occurs. Apparently, data gathered from adjacent nodes have the spatial redundancy; that is, nodes in adjacent regions may collect and transmit similar data (e.g., the humidity distributions around a tea tree). Without proper aggregation operations in data collection and transmission processes, such a significant amount of data can easily result in imbalance of energy consumption, thereby shortening the lifetime of whole network. Therefore, we should reduce the amount of data transferred in the network, improve the efficiency of data transmission, and save node energy when researching the 6LoWPAN network.

Contributions. The design of 6LoWPAN protocol stack does not take data aggregation into account, resulting in data redundancy. Data aggregation [4–6] can save the bandwidth when sending and receiving data and can improve the efficiency of data collection of the network as well. But it will lead to transmission delay. In this paper, we focus on relatively dense 6LoWPAN networks, and the main contributions of this paper include the following.

- (1) We distinguish the property of data, do some aggregation operation for those non-real-time data, and try to reduce the transmission delay.
- (2) We introduce the data aggregation function to the adaptation layer of 6LoWPAN protocol stack.

- (3) We propose a dynamic data aggregation mechanism assisted by location information, which can aggregate the data adaptively according to the network status.

Through this mechanism, we can obviously reduce the amount of data, improve the real time performance of network, and make an appropriate trade-off between energy consumption and transmission delay.

The remainder of the paper is organized as follows. In Section 2, we describe briefly the relevant background knowledge of 6LoWPAN. And our data aggregation mechanism is detailed in Section 3. Section 4 analyzes the simulation results of the proposed mechanism. Finally, the conclusions are presented in Section 5.

2. Related Work

In order to reduce the redundancy and shrink the total amount of data in sensor networks, data aggregation has been well studied by numerous researchers in recent years [7–9]. In [10], data aggregation methods are classified according to the level of sampled data, that is, data compression aggregation [11, 12], in-network aggregation [13, 14], and representative aggregation [10, 15]. Applying any of these data aggregation methods can reduce the redundancy of data in sensor networks to some extent; however, each of them has certain limitations. Data compression aggregation is expected to play an important role in data aggregation techniques, especially with the development of compressed sensing recently. However, this type has postponed the development as it is too complex to operate.

Guo et al. [13] proposed a novel location-based in-network data aggregation technique. This technique is designed to explore the data aggregation process in the cluster as well as the location information assisting process. By selecting the data transmission nodes which have the same region location information but different readings compared with the agent nodes, the technique can effectively reduce the in-cluster spatial redundancy. However, this inner-network data aggregation results in a larger interval time to send a reply to the service requestor and cannot satisfy real-time requirement.

Another in-network data aggregation is described in [14], which allows nodes to form a core structure. Core nodes are elected based on the Minimum Dominating Set (MDS). By using MDS, the number of core nodes can reach minimum to cover all parts of the non-core-nodes in the network. The aggregation algorithm relies on the Shortest Path Tree (SPT) for transmitting the aggregated data hop-by-hop to sink node. It reduces the delivery cost inside the network and promotes the bandwidth utilization in 6LoWPAN.

There are some other representative aggregation strategies proposed recently with the coverage model and location information to achieve energy saving [10, 15]. In [15], Shakya et al. addressed the problem of energy-efficient data collection process in the existence of spatial correlation in relatively dense wireless networks. They mainly focus on location-based correlation characteristics among sensor nodes. The spatial correlation model needs precise nodes positioning. In some harsh environment, especially, the correctness of spatial

correlation among nodes would be greatly debased, thereby dramatically impacting the performance of the representative aggregation algorithm.

Yuan et al. [10] explored the effect of imprecise nodes positioning on spatial correlation models and provided a data-density based correlation degree algorithm to solve the above problem. According to this correlation degree algorithm, a hierarchical structure is established to differentiate correlation of nodes. Nevertheless, the relative error between the representative and observed value of the sensor nodes would reduce to an unacceptable level, because the impact of residual energy of representative nodes is neglected.

In summary, there have been lots of works on data aggregation, including time scheduling, novel aggregation operating, and routing plans. However, it is equally important to focus on other objectives such as in-node aggregation, and existing aggregation algorithms almost run on the intermediate node. Furthermore, the IPv6 header is usually larger than the data in it, and it is necessary to add the data aggregation mechanism to the 6LoWPAN protocol for improving the transmission efficiency.

3. System Model

3.1. 6LoWPAN Protocol Stack. 6LoWPAN requires that its MAC layer and physical layer must comply with IEEE 802.15.4 standard, and its network layer must use IPv6 protocol. Our previous works [16] have demonstrated that adding an adaptation layer between IPv6 network layer and IEEE 802.15.4 MAC layer is an efficient and accurate method to run IPv6 on IEEE 802.15.4 standard, which is the most important layer of 6LoWPAN protocol stack. The basic functions of 6LoWPAN, such as header compression, multicast support, fragmentation and reconstruction, address distribution, and network topology construction, are all implemented in the adaptation layer [17]. The adaptation layer can provide medium access for the upper IPv6 layer and invoke the lower layer functions such as network construction, topology control, and MAC routing.

Header Compression. The adaptation layer of 6LoWPAN compresses the header and subsequent header of IPv6 packets by using context-based method and stateless compression methods, which could improve the transmission efficiency and reduce the energy consumption of nodes. Without using security features, the maximum transferable length of IEEE 802.15.4 MAC layer is 102 B. However, the basic length of the IPv6 header is 40 B. Apart from the adaptation header and transport header (such as UDP), it will only be available for 50 B to be used by upper application data. In order to satisfy the maximum transmission unit (MTU) of IPv6 under the IEEE 802.15.4 MAC layer, 6LoWPAN has been expanded and improved in [18] based on early definition in [19].

Multicast Support. Multicast technology plays a very important role in IPv6 protocol. Many functions in Neighbor Discovery Protocol (NDP) rely on it. IEEE 802.15.4 MAC layer only provides limited broadcast. Therefore, the multicast

function of adaptation layer can be realized by using controllable broadcast flooding.

Fragmentation and Restructuring. IPv6 stipulates that the MTU size is no less than 1280 B [20]. Those links that do not support this MTU must realize the fragmentation and restructuring function, so that the adaptation layer of 6LoWPAN implements this function to cope with the frame whose length is bigger than MTU.

MAC Layer Routing. Due to the absence of multihop routing in MAC layer, adaptation layer defines two kinds of routing mechanism based on the address assignment mechanism, that is, tree routing and network routing.

Network Topology Construction and Address Assignment. IEEE 802.15.4 standards define PHY layer and MAC layer in detail. MAC layer provides abundant primitives, including network maintenance and channel scanning, while MAC layer is not responsible for the construction and maintenance of network topology by calling these primitives. Therefore, these works will be achieved by adaptation layer. Each device in 6LoWPAN network has a EUI-64 address which is too large and will increase frame length. Adaptation layer implements a 16-bit address allocation mechanism to identify each node in the network.

6LoWPAN network uses multihop to transmit data. The amount of data will increase because of the growing of running time and network size, and the transmission efficiency will decline at the same time. These will lead to large amount of transport controls, as well as the increasing of channel competition and network congestion. Therefore, reducing the amount of transmission control messages, as well as the channel competition and network congestion, is an important issue.

3.2. Network Architecture. Based on the distribution of sensor nodes depicted in Figure 1, we establish a 6LoWPAN network for agriculture surveillance, as shown in Figure 2. Suppose there are N sensor nodes that are randomly deployed in 9 tea fields. Node types in this network include common sensor nodes, aggregation nodes, and a sink node. When sink node receives a service request, it uses a spanning tree algorithm for data aggregation to generate a tree topology. This spanning tree algorithm uses sink node as the root node and selects multihop paths to the aggregation node. Then common sensor nodes start to sample sensory information from the monitored area (e.g., temperature, humidity, and sunlight).

Our network architecture could be represented as an undirected graph $G(N, L)$ that is composed of a set of nodes N and a set of links L . For each node $i \in N$, F_i represents the set of neighbor nodes, i could communicate with any node in F_i , where $\{(i, j) \in L \mid i, j \in N, j \in F_i\}$. C denotes the set of common sensor nodes, A is the set of aggregation nodes, and S stands for the unique sink node, where $N = C \cup A \cup S$. Since our data aggregation model only involves sensory data collecting and transmitting, we pay no attention to the mobility of nodes; that is, the network is a static network.

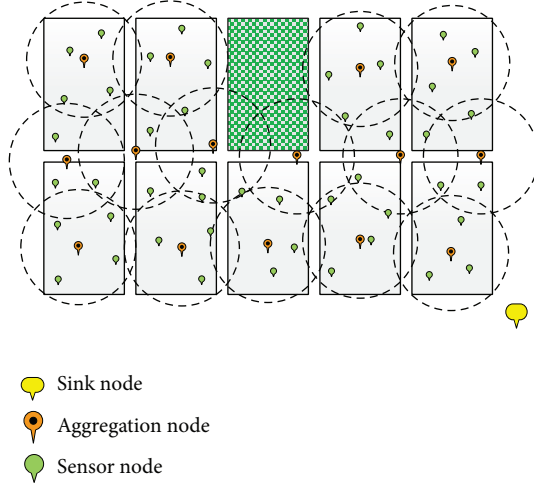


FIGURE 2: Data monitoring diagram in 6LoWPAN.

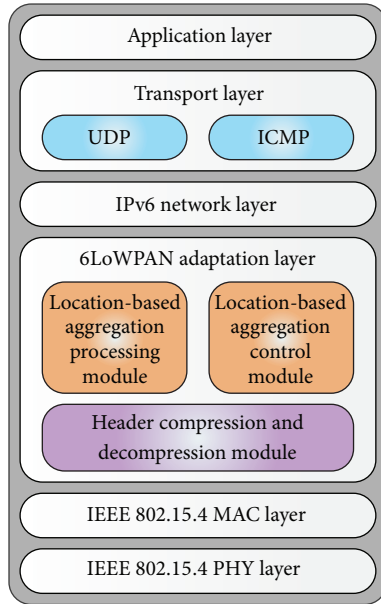


FIGURE 3: Aggregation architecture.

4. Data-Aggregation-Based 6LoWPAN Stack

4.1. Data Aggregation Model. In order to reduce the amount of control messages, as well as the costs of channel listening and competition, we add an aggregation model to the adaptation layer of 6LoWPAN stack. The architecture of aggregation model is shown in Figure 3. It consists of two submodules: location-based aggregation processing module and location-based aggregation control module. The location-based aggregation processing module includes an input queue, a data aggregation unit, and an output queue. Aggregation parameters include aggregation-condition (**P**), aggregation-content (**C**), time-granularity (**T**), and aggregation-degree (**D**). By setting these parameters, the aggregation control module could adjust the data aggregation dynamically.

The work flow of aggregation model is as follows. Firstly, the adaptation layer receives IPv6 packet from the network layer and pushes it into input queue, and then the IPv6 packet will be classified into real-time data and non-real-time data. If it is real-time data, it will be sent directly to the output queue; if not, a certain delay does not have a significant impact. Therefore, the packet will be sent to the aggregation unit. After that, it will also be sent to the output queue.

4.2. Aggregation Strategy. In this paper, we use the four-tuple $\langle P, C, T, D \rangle$ to represent the aggregation strategy.

P is the aggregation condition which consists of traffic class and destination location information. Traffic class is used to identify real-time data or any data requiring to be handled specially. Destination location information is utilized to identify the packets which are from the same region of interest.

C is the content of aggregation which is from the network layer to the adaptation layer.

T is a fixed time granularity. We set this parameter according to

$$T = \frac{\alpha M}{R}, \quad (1)$$

where M is the actual monitoring time of application, R is the accuracy requirement of application, and the factor α is related to specific application, valued from 0 to 1. If the application has a longer M or a lower R , the T will be bigger, and the aggregation time and the amount of aggregated data will increase. Otherwise, the T will be smaller, and the aggregation time will decrease, as well as the amount of aggregated data.

D is the aggregate granularity. It is the number of the packets that can be aggregated together in certain aggregation condition. The definition is shown as follows:

$$D_0 = \frac{\sum_{d(i,A) \leq \eta} N_r}{\sum_{d(i,A) \leq \eta} N_s},$$

$$D_t = \begin{cases} D_{t-1} \left(1 - \frac{T}{QD}\right) & \frac{T}{QD} > \zeta \\ D_{t-1} \left(1 + \frac{T}{QD}\right) & \frac{T}{QD} \leq \frac{1}{\zeta} \end{cases} \quad t > 0, \quad (2)$$

where N_r represents the total number of received packets from region of interest; that is, $d(i, A) \leq \eta$. $d(i, A)$ represents the distance between node i and the aggregation node A ; η is the transmission range of A . N_s means the number of packets involving aggregation from the region of interest, and QD is output queuing delay. If **D** is bigger, we can transmit more data in one packet header, which will reduce the header overhead and the data redundancy, as well as improving the channel efficiency. And ζ denotes the threshold of **D**.

In this paper, we use the output queuing delay (QD) to represent MAC layer delay and adopt loop control method to adjust **D** dynamically which will reduce the collision probability and waiting time in wireless channels. We can get real-time queuing delay by monitoring the output queue of the adaptation layer and then use it to adjust **D**. In the process of aggregation, if multiple packets find the same **P**, the data

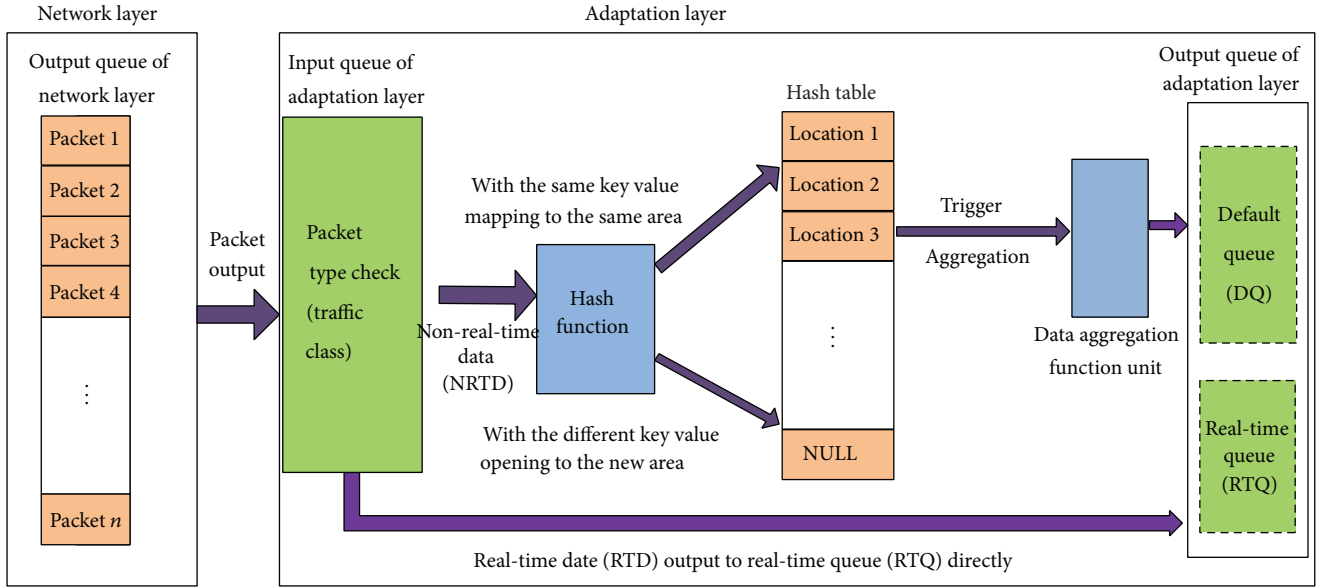


FIGURE 4: Data aggregation based on the aggregation conditions block diagram.

aggregation unit will aggregate C in a certain D value and output the result to the output queue within T .

4.3. Location-Based Dynamic Aggregation Algorithm (LDAA). 6LoWPAN networks are composed of massive nodes and have efficient mechanisms for encapsulation and header compression. If we forward collected data directly, this will contribute to reducing data transmission efficiency, wasting wireless bandwidth, and increasing the conflict probability as well as the energy consumption. Due to the high-density deployment and high traffic load in our networks, congestion can occur due to burst traffic in some applications. In the case of an emergency (e.g., a tea tree fire) scenario, if the channel is already congested, then these crucial messages (e.g., messages of fire) could not access the channel successfully, which causes crucial messages loss or higher delays. Thus, data from different applications need to have different priorities. To solve the problems above, we propose the Location-Based Dynamic Aggregation Algorithm (LDAA) in this paper.

Data aggregation can effectively reduce the amount of data in the 6LoWPAN networks, but it can still lead to some delay costs. According to the Traffic Class field of the packet, we divide data into high-priority real-time data (RTD) and low-priority non-real-time data (NRTD). The output queue of adaptation layer contains two independent subqueues; one is the Real-Time Queue (RTQ) for RTD and the other one is the Default Queue (DQ) for NRTD. Each RTD will be sent to the MAC layer by RTQ in real-time speed; it ensures that RTD will not encounter unexpected queue delays and congestion. On the other hand, NRTD in DQ will do the best on data output; it probably experiences a significant delay and high levels of congestion, as shown in Figure 4.

The packets need to aggregate according to their destination location information. First, we should classify the input data into different groups. When the number of the packets

with the same destination location information reaches the aggregation-degree D , these packets will be sent to the aggregation processing module. Thus, the storage structure we adopted must be suited for this function. Because Hash-Table has a characteristic that could be mapped quickly from key to value, the LDAA algorithm maps the packet into a Hash-Table using the geographical position of the destination node as the key. The structure of the Hash-Table shows, in Figure 4, that each element of the Hash-Table is a packet list that has the saved destination location information. Within the presetting time-granularity T , if the length of the list reaches the aggregation-degree D , we will output the whole list into aggregation processing module and send the aggregated result to the DQ for MAC layer transmission.

To avoid a long delay for reaching the aggregation-degree D , we start a timer when the packet list is constructed. If the timer ends, we output the list no matter whether reaching D or not, and then restart the timer when a new packet comes into the list.

Algorithm 1 shows the pseudocode of LDAA algorithm.

5. Simulation Results and Analysis

5.1. Simulation Parameters Setting. The simulation experiments are conducted on MyEclipse 8.5 (with JAVA programming language) to evaluate the performance of Location-Based Aggregation Algorithm. 100 sensor nodes are densely deployed with the maximum transmission distance 30 m in a region of 100 m \times 100 m. We assume that the packet length is fixed at 4000 bits. The initial energy and energy consumption per unit of each node are set as 1 J and 50 nJ, respectively. Each node has the same packet sending interval which represents the time between two consecutive packets. Other simulation parameters are listed in Table 1.

Input: queue of adaptation layer, $\langle P, C, T, D \rangle$, QD, ζ , where $t \geq 0$.

Output: queue of adaptation layer.

Event: Edge routers request data to node S in region of interest.

/ Steps */*

- (1) Start a timer to avoid a long delay for reaching D_0 ;
- (2) The adaptation layer receives the packets from the output queue in the network layer;
- (3) Push the received packets into the input queue;
- (4) Check the Traffic Class in P ;
- (5) **if** It is RTD **then**
- (6) Go to (10);
- (7) **else**
- (8) Go to (12);
- (9) **end if**
- (10) The aggregation control module outputs the RTD to the RTQ directly;
- (11) Go to (43);
- (12) The aggregation control module maps NRTD packet into Hash-Table based on the destination location information in P ;
- (13) **if** The destination location information exists in the Hash-Table **then**
- (14) Go to (18);
- (15) **else**
- (16) Go to (25);
- (17) **end if**
- (18) The aggregation control module adds the packet to the packet list with the same location information;
- (19) **if** The length of the list is empty **then**
- (20) Restart the timer related to the list;
- (21) **else**
- (22) Calculate the length of the packet list;
- (23) Go to (28);
- (24) **end if**
- (25) The aggregation control module constructs a new packet list in the Hash-Table for the new destination location information;
- (26) Add the packet into the list;
- (27) Start a new timer for this list;
- (28) **if** The timer of the packets list is end **then**
- (29) Go to (43);
- (30) **else**
- (31) Go to (33);
- (32) **end if**
- (33) **if** $T/QD > \zeta$ **then**
- (34) $D_t = D_{t-1}(1 - T/QD)$;
- (35) **end if**
- (36) **if** $T/QD \leq 1/\zeta$ **then**
- (37) $D_t = D_{t-1}(1 + T/QD)$;
- (38) **end if**
- (39) **if** The length of the packets list equals with D_t **then**
- (40) The aggregation control module outputs the list to the aggregation processing module;
- (41) **end if**
- (42) The aggregation processing module aggregates multiple packets and outputs the results into DQ;
- (43) The adaptation layer calls the sending module of the MAC layer to send the data.
- (44) **return**

ALGORITHM 1: LDAA algorithm.

The main idea of Dynamic Location-Based Aggregation Algorithm (LDAA) is to adjust the aggregation degree D dynamically based on the traffic load (as shown in (2)). In this paper, our scheme is targeted for relatively dense 6LoWPAN networks which have high traffic load. In order to fully analyze the performance of LDAA, we compare it with NOA (non-data aggregation algorithm) and FIXA (fixed aggregation degree data aggregation algorithm) on two measurable metrics. One is the end-to-end delay of packet,

and the other is the energy consumption calculated by First Order Radio Model [21, 22].

5.2. End-to-End Delay. The end-to-end delay is the main factor of the network real-time performance. Therefore, under the same network load condition, the shorter the end-to-end delay is, the higher the network real-time performance will be.

TABLE 1: Simulation parameters.

Parameter	Value
Simulation area	100 m × 100 m
Sink node coordinate	(30, 10)
Total number of nodes	100
Initial energy of each node	1 J
Energy consumption per unit	50 nJ/bit
Maximum transmission distance of each node	30 m
Packet length	4000 bits
Packets sending interval	100 ms–170 ms

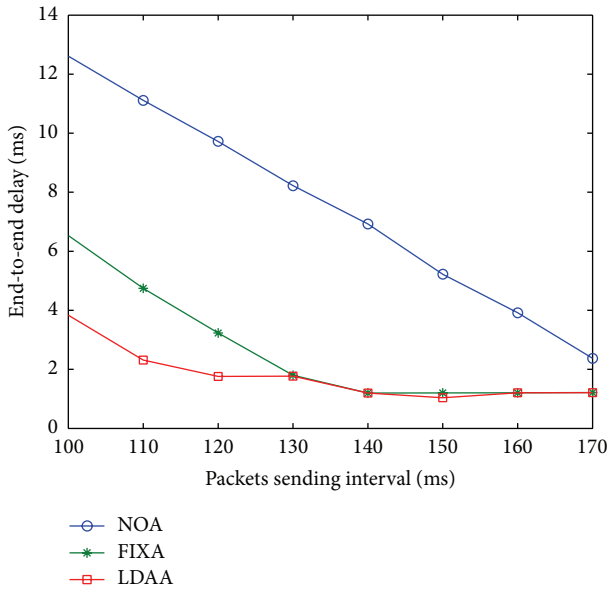


FIGURE 5: Packets sending interval compared with the end-to-end delay.

Figure 5 depicts the end-to-end delay under different packets sending intervals of the LDAA, NOA, and FIXA algorithms. It is noticed that the end-to-end delay in NOA is longer than that in FIXA and LDAA. This is because NOA has no data aggregation method to reduce the amount of transmitted data. When the packets sending interval changes between 100 and 130, the LDAA would decrease the aggregation degree to meet the decreased network load. But the aggregation degree of FIXA is fixed; it cannot adapt to the dynamic load network. The LDAA and FIXA algorithms have the same end-to-end delay for packets sending interval between 130 and 170, and the aggregation degree of them is almost similar.

Another more detailed end-to-end delay comparison of LDAA and FIXA algorithms with different aggregation degrees is shown in Figure 6. We can find that the performance of FIXA with 4 degrees is better than that with 2 and 3 degrees when the packets sending interval is less than 110. Between 110 and 130, the end-to-end delay of 3 degrees is the lowest one. Conversely, 2 degrees outperform 3 and 4 degrees in the other case. Obviously, it is observed that LDAA remains lowest as compared with FIXA with different aggregation

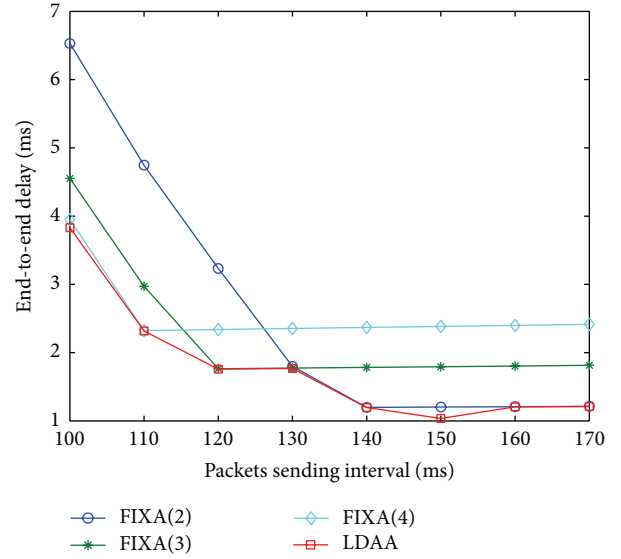


FIGURE 6: The end-to-end delay comparison of LDAA and FIXA with different aggregation degrees.

degrees. This is because the LDAA algorithm can adjust the aggregation degree dynamically according to the changes of network load. There is no adaptive adjustment mechanism of aggregation degree in FIXA. Consequently, the LDAA algorithm is superior to the NOA and FIXA algorithms on the aspect of end-to-end delay. Moreover, LDAA is more suitable for the networks with dynamic load and could promote the network bandwidth utilization and shorten the transmission delay.

5.3. Energy Consumption. The relationship between packets sending interval and the energy consumption is shown in Figure 7. We can notice that NOA and FIXA have no obvious changes in energy consumption when the sending interval of packet increases. And the energy consumption of the LDAA algorithm is better than the NOA and FIXA algorithm. This is because the LDAA algorithm adopts data aggregation operation, which can efficiently reduce the amount of data, and hence the transmission power is lower. As the number of packets sending interval increases, the LDAA algorithm has the similar energy consumption to the FIXA. Because of the dynamic feature of LDAA algorithm, LDAA can adjust the appropriate aggregation degree to make its energy consumption per packet lower than NOA and FIXA.

As we can see from Figure 8, the change in the energy consumption of FIXA algorithm with different aggregation degrees is significant as the packets sending interval increases. The more the aggregation degree grows, the lower the energy consumption will get. When the aggregation degree is 3, the energy consumption decreases about 20% more than the degree of 2. When the aggregation degree is 4, the energy consumption decreases about 28% more than the degree of 2. The higher the aggregation degree is, the more the data packets which share the same header will be and the lower the energy consumption per packet will get. Similarly to

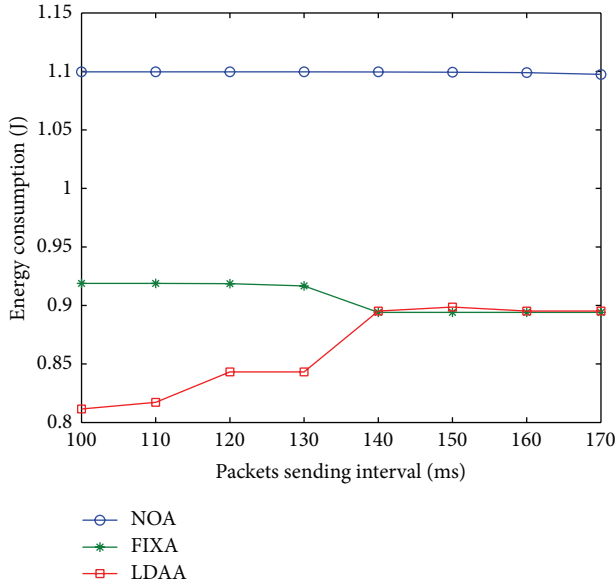


FIGURE 7: Comparison of different algorithms for energy consumption in the network.

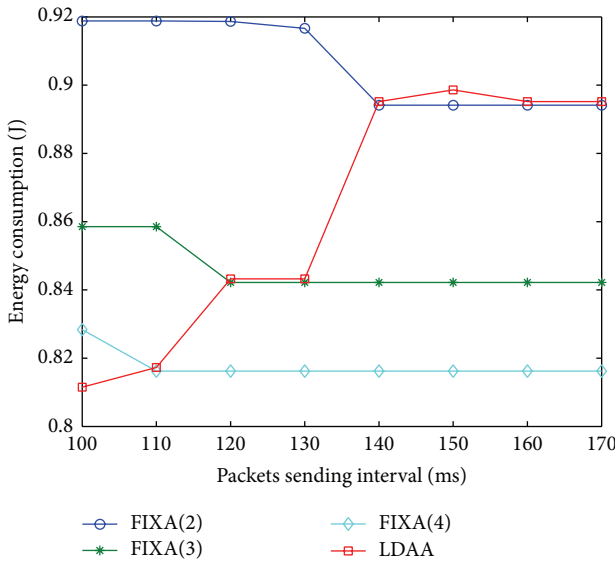


FIGURE 8: LDAA with different degrees of polymerization of FIXA energy consumption compared.

the situation as shown in Figure 6, LDAA can make intelligent decisions on energy consumption based on the adaptive adjustment mechanism of aggregation degree, thus reducing the energy consumption efficiently.

Based on the analysis of the end-to-end delay and energy consumption of these three algorithms, we conclude that the network performance would not keep improving with the aggregation degree increases. With the network load increases, the algorithm could adjust the aggregation mechanisms dynamically to provide better latency and energy consumption. Therefore, LDAA is targeted for the high-loading network.

6. Conclusions

In this paper, we have designed an aggregation model in the adaptation layer of 6LoWPAN protocol stack to reduce the control overhead and the MAC layer delay as well as the spatial redundancy. We propose the Location-Based Dynamic Aggregation Algorithm (LDAA) based on aggregation conditions. LDAA could obtain the network load in real time by monitoring the MAC layer delay and adjust the aggregation degree dynamically by monitoring the network load. This algorithm divides IPv6 packets into different priorities based on the aggregation strategy. For real-time data, LDAA would send it directly to the MAC layer without executing any aggregation. For non-real-time data, LDAA would aggregate it according to the aggregation degree. Clearly, the algorithm can not only effectively reduce the spatial redundancy of data and the energy consumption of 6LoWPAN networks, but also provide better real-time guarantees for data in networks to meet the requirements of the 6LoWPAN applications.

Conflict of Interests

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

Acknowledgments

This work was partially supported by National Key Technology R&D Program (2012BAD35B06); Program for New Century Excellent Talents in University (NCET-12-0164); National Natural Science Foundation of China (61370094); Natural Science Foundation of Hunan (13JJ1014).

References

- [1] M. Chui, M. Löffler, and R. Roberts, "The internet of things," *McKinsey Quarterly*, no. 2, pp. 1–9, 2010.
- [2] S. A. Chaudhry, A. H. Akbar, and K. Ki-Hyung, "On the interplay of service proximity and ubiquity," *IEICE Transactions on Communications*, vol. 90, no. 12, pp. 3470–3479, 2007.
- [3] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*, vol. 43, John Wiley & Sons, New York, NY, USA, 2011.
- [4] R. Rajagopalan and P. K. Varshney, "Data-aggregation techniques in sensor networks: a survey," *IEEE Communications Surveys and Tutorials*, vol. 8, no. 4, pp. 48–63, 2006.
- [5] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 14, no. 2, pp. 70–87, 2007.
- [6] V. Pandey, "A review on data aggregation techniques in wireless sensor network," *Journal of Electronic and Electrical Engineering*, vol. 1, no. 2, 2010.
- [7] M. Bagaa, Y. Challal, A. Ouadjaout, N. Lasla, and N. Badache, "Efficient data aggregation with in-network integrity control for wsn," *Journal of Parallel and Distributed Computing*, vol. 72, no. 10, pp. 1157–1170, 2012.
- [8] L. Zhu, Z. Yang, J. Xue, and C. Guo, "An efficient confidentiality and integrity preserving aggregation protocol in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 565480, 8 pages, 2014.
- [9] Y. Choi and I. Doh, "Hierarchical aggregation of uncertain sensor data for m2m wireless sensor network using reinforcement

- learning,” *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 535707, 9 pages, 2014.
- [10] F. Yuan, Y. Zhan, and Y. Wang, “Data density correlation degree clustering method for data aggregation in WSN,” *IEEE Sensors Journal*, vol. 14, no. 4, pp. 1089–1098, 2014.
 - [11] L. Xiang, J. Luo, and C. Rosenberg, “Compressed data aggregation: energy-efficient and high-fidelity data collection,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1722–1735, 2013.
 - [12] G. Cao, P. Jung, S. Stańczak, and F. Yu, “Low cost error correction for multi-hop data aggregation using compressed sensing,” *IEICE Transactions on Information and Systems*, vol. 97, no. 2, pp. 331–334, 2014.
 - [13] J. Guo, H. Zhang, and W. Chen, “Location-based inner-cluster data aggregation for wireless sensor networks,” *AASRI Procedia*, vol. 3, pp. 523–527, 2012.
 - [14] S. J. Park and R. Sivakumar, “Energy efficient correlated data aggregation for wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 4, no. 1, pp. 13–27, 2008.
 - [15] R. K. Shakya, Y. N. Singh, and N. K. Verma, “A novel spatial correlation model for wireless sensor network applications,” in *Proceedings of the 9th IEEE and IFIP International Conference on Wireless and Optical Communications Networks (WOCN '12)*, pp. 1–6, Indore, India, September 2012.
 - [16] J. Luo, C. Pan, and P. W. Wang, “Data aggregation method based on 6lowpan protocol stack,” CN Patent App, CN 201,310,233,531, 2013.
 - [17] X. Ma and W. Luo, “The analysis of 6lowpan technology,” in *Proceedings of the Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 1, pp. 963–966, Wuhan, China, December 2008.
 - [18] J. Hui and P. Thubert, “Compression format for IPv6 datagrams in 6LoWPAN networks. draft-ietf-6lowpan-hc-13 (work in progress),” 2010, <https://tools.ietf.org/html/draft-ietf-6lowpan-hc-13>.
 - [19] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, “Transmission of ipv6 packets over IEEE 802.15.4 networks,” Internet Proposed Standard RFC 4944, 2007.
 - [20] S. Deering and R. Hinden, “Internet protocol, version 6 (ipv6) specification,” RFC 1883, IETF, 1998.
 - [21] M. Bhardwaj, T. Garnett, and A. P. Chandrakasan, “Upper bounds on the lifetime of sensor networks,” in *Proceedings of the International Conference on Communications (ICC '01)*, vol. 3, pp. 785–790, June 2000.
 - [22] R. Min, M. Bhardwaj, N. Ickes, A. Wang, and A. Chandrakasan, “The hardware and the network: total-system strategies for power aware wireless microsensors,” in *Proceedings of the IEEE CAS Workshop on Wireless Communications and Networking*, pp. 36–42, IEEE, Pasadena, Calif, USA, September 2002.

