



## Article

# Adaptive Aggregation Routing to Reduce Delay for Multi-Layer Wireless Sensor Networks

Xujing Li <sup>1</sup>, Anfeng Liu <sup>1,2</sup> , Mande Xie <sup>3,\*</sup>, Neal N. Xiong <sup>4</sup> , Zhiwen Zeng <sup>1</sup> and Zhiping Cai <sup>5</sup>

<sup>1</sup> School of Information Science and Engineering, Central South University, Changsha 410083, China; xujingli@csu.edu.cn (X.L.); anfengliu@mail.csu.edu.cn (A.L.); zengzhiwen@mail.csu.edu.cn (Z.Z.)

<sup>2</sup> The State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China

<sup>3</sup> School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

<sup>4</sup> Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA; xionгнаixue@gmail.com

<sup>5</sup> Department of Network Engineering, School of Computer, National University of Defense Technology, Changsha 410073, China; zpcai@nudt.edu.cn

\* Correspondence: xiemd@zjgsu.edu.cn; Tel.: +86-731-8887-9628

Received: 30 March 2018; Accepted: 12 April 2018; Published: 16 April 2018



**Abstract:** The quality of service (QoS) regarding delay, lifetime and reliability is the key to the application of wireless sensor networks (WSNs). Data aggregation is a method to effectively reduce the data transmission volume and improve the lifetime of a network. In the previous study, a common strategy required that data wait in the queue. When the length of the queue is greater than or equal to the predetermined aggregation threshold ( $N_t$ ) or the waiting time is equal to the aggregation timer ( $T_t$ ), data are forwarded at the expense of an increase in the delay. The primary contributions of the proposed Adaptive Aggregation Routing (AAR) scheme are the following: (a) the senders select the forwarding node dynamically according to the length of the data queue, which effectively reduces the delay. In the AAR scheme, the senders send data to the nodes with a long data queue. The advantages are that first, the nodes with a long data queue need a small amount of data to perform aggregation; therefore, the transmitted data can be fully utilized to make these nodes aggregate. Second, this scheme balances the aggregating and data sending load; thus, the lifetime increases. (b) An improved AAR scheme is proposed to improve the QoS. The aggregation deadline ( $T_t$ ) and the aggregation threshold ( $N_t$ ) are dynamically changed in the network. In WSNs, nodes far from the sink have residual energy because these nodes transmit less data than the other nodes. In the improved AAR scheme, the nodes far from the sink have a small value of  $T_t$  and  $N_t$  to reduce delay, and the nodes near the sink are set to a large value of  $T_t$  and  $N_t$  to reduce energy consumption. Thus, the end to end delay is reduced, a longer lifetime is achieved, and the residual energy is fully used. Simulation results demonstrate that compared with the previous scheme, the performance of the AAR scheme is improved. This scheme reduces the delay by 14.91%, improves the lifetime by 30.91%, and increases energy efficiency by 76.40%.

**Keywords:** wireless sensor networks; data aggregation; aggregation delay; aggregation ratio; energy efficiency

## 1. Introduction

In the past several decades, there have been a variety of developments in wireless sensor networks (WSNs) in both academia and industry [1–5]. A wireless sensor network is a set of distributed hardware-constrained wireless devices that monitor a targeted area [6–11]. Wireless sensor

networks are widely used in a variety of applications such as the environmental monitoring and urban health monitoring [12–14], healthcare [2,15], vehicular communication networks [16], cyber-physical cloud systems [17], multi-channel cognitive radio networks [18,19], crowdsourcing networks [20–22], and social networks [23,24]. Because of the growing demand for wireless sensor applications in various aspects, quality of service (QoS) is an important issue in wireless sensor applications [25–29].

For the QoS of wireless sensor applications, energy efficiency [5,11,17,30,31] and delay [8,9,27,29] are two of the most significant indicators. Since wireless sensor nodes are battery-powered, their energy is limited. Thus, improving energy efficiency [14,15,23,28,32–34] is an important research area for WSNs. There are many studies regarding energy efficiency, which include optimization in a single layer and joint optimization in multiple layers. For instance, optimization methods in the MAC layer [2,11,26], network layer and application layer, and cross-layer optimization methods have been proposed. Methods that address the MAC layer include adjusting the transmission power of the nodes [2,11,26], selecting optimized MAC parameters [2,11,26], and the dynamic duty cycle [35]. The network routing optimization method in the network layer [7,32] and selecting the optimal size of data packets in the application layer [2] are effective methods, as well. Among these methods, data aggregation is an effective and widely studied method [7,29,36–38]. Since there is a correlation between sensing data, the amount of data packets after aggregation is less than the original amount when two or more data packets aggregate, thereby reducing the number of network data packets [36–38]. Receiving and sending data packets consumes the most energy in sensing nodes; therefore, reducing the number of data packets to be transmitted can reduce energy consumption. Therefore, improving the efficiency of data aggregation is an effective approach to improving network energy efficiency. An important principle for improving the data aggregation efficiency requires data packets to encounter as many other data packets possible during the process of data collection to increase the probability of data aggregation [29,36,38]. In many data aggregation methods, when a node receives data packets, it does not send them immediately but holds them for a period of time [29] to increase the number of data packets waiting in the aggregation queue. When the number of data packets in a node is the same or greater than the predetermined threshold or the aggregation timer (AT) expires, the data packets are aggregated and transmitted. However, this effect causes an increase in the delay. With a large AT value, data aggregation works well, the data aggregation energy consumption decreases, but the delay significantly increases. The relationship between the delay, the length of the data queue and the AT has been confirmed by researchers. The average aggregation delay linearly increases as the aggregation timer or the aggregation threshold increases [29].

Although researchers have confirmed the optimization relationship, improving the QoS of the wireless sensor network remains a challenging issue [2]. The current research in optimizing aggregation delay is confined to networks with a star topology [29]. No study has been proposed to improve the QoS for general planar WSNs. The previous conclusions may provide guidance for the optimization of general planar WSNs. Planar WSNs are more complicated than star WSNs. The planar wireless sensor network is abstracted as a tree network in many studies. In this kind of network, the root of the network is the sink [39,40]. Other nodes in the network receive data from their child nodes, except for leaf nodes. There may be multiple children under a parent node, and each node may have multiple parents [26,41]. There are diverse routing paths for a node to transmit its aggregation queues. It dynamically changes the structure of the network and delay is affected. These influences interfere with each other. Thus, proposing an optimization strategy for data aggregation in a planar network is very challenging.

In this paper, an adaptive aggregation routing (AAR) scheme is proposed to reduce the delay and improve the lifetime for multi-layer wireless sensor networks. The main innovations of this paper are as follows:

- (1) An adaptive aggregation routing (AAR) scheme is proposed to reduce delay and improve the lifetime for multi-layer wireless sensor networks. The core of this scheme is the node assignment algorithm (NAAL) that we proposed. This algorithm addressed the assignment of data queues

in two adjacent layers of the network. According to the state of the data queue in the nodes, this algorithm selects nodes with a long data queue in the upper layer and sets the priority according to the length of the data queue. These nodes are guaranteed to receive sufficient data to aggregate, while other nodes which have no data to send are put to sleep in this process to save energy. Therefore, the frequency of data aggregation increases and the total delay decreases. Simulation results illustrate that the AAR scheme reduces the delay by 14.91% and improves the lifetime by 30.91% compared to other common schemes.

- (2) Based on the AAR scheme, an improved optimization method is proposed in this paper, which improves the QoS. The main idea of the improved ARR scheme is as follows: the data in the nodes far from the sink use more hops to be transmitted to the sink, and there is delay in every hop. Consequently, there is noticeable delay caused during the process of transmitting the data from the distant nodes to the sink. In WSNs, nodes far away from the sink transmit less data than the other nodes. Therefore, more energy remains in these nodes. In the improved AAR scheme, the nodes far away from the sink have a small aggregation deadline ( $T_t$ ) value and queue length threshold ( $N_t$ ) to improve the frequency of sending the aggregation queues. The nodes near the sink are set to a large  $T_t$  value and  $N_t$  value to reduce energy consumption. The total delay is reduced, high lifetime is achieved, and the residual energy is fully used. Based on the simulations, the improved AAR scheme increases the energy efficiency by 76.40%.
- (3) To evaluate its effectiveness, we conducted extensive simulations in a variety of network environments. The results indicate that the scheme we proposed performs better than other common networks.

The rest of this paper is organized as follows: in Section 2, related works are reviewed. The system model and problem statement are described in Section 3. In Section 4, design details on AAR scheme are presented. Section 5 is experiment results and comparisons for AAR scheme. Finally, Section 6 provides the study's conclusions.

## 2. Related Work

This section presents previous work related to optimizing energy efficiency and delay in wireless sensor networks (WSNs).

### 2.1. Research on Data Aggregation

Data aggregation is a major operation in WSNs [7,29,36–38]. Sensor nodes are deployed in a certain area, and the sensing data are correlated in time and space. As a result, nodes aggregate data to a smaller size. Different research methods have been adopted according to different data aggregation ratios. A useful and widely researched data aggregation scenario is that any number of packets are aggregated into a single packet. For example, calculating the average temperature or maximum temperature in the monitoring area. In this case, multiple data packets are aggregated into one data packet. This type of data aggregation problem is often abstracted as a convergecast problem [42,43]. The key character the process of collecting data is that each node only receives data in the data collection phase, and only aggregates data packets into one data packet which is sent in the data transmission phase. In the data transmission phase, regardless of the number of data packets transmitted to the nodes, no data are received by the nodes. Several convergecast algorithms [42,43] have been proposed for wireless sensor networks. Most studies divide this issue into two parts. The first part is a logical tree construction, followed by the scheduling of transmissions along the constructed tree.

Convergecast is a special case of data aggregation. The general case of data aggregation is that sensors aggregate data using a certain percentage. In the data transmission process from the source to the sink node, with continuous data aggregation, the data volume of packets increases [3,7,29,36,38]. To reduce the number of data packets, it is necessary to aggregate as many packets as possible.

Villas et al. [38] proposed a data aggregation method named DRINA which improves the probability of packets routing along the same path. This method improves the probability of encountering data packets, and data are aggregated more effectively; therefore, the data size decreases. The strategy is described as follows [38]: the minimum hop routing strategy is adopted, that is, each node chooses the node with the smallest hop to the sink as the next hop for data forwarding. The formation of the minimum hop to the sink is not the same as the previous strategy. When a node has data packets to send to the sink, it chooses a routing path according to a minimum hop routing strategy. Then, the number of node hops on this path is set to 0. With hop spreading, the data in the nodes in this path route to the sink along this path. As a result, the probability of packets routing along the same path increases.

The method proposed by Kim et al. [29] to improve data aggregation performance is different from the methods mentioned above. When a node receives data packets, it does not send them immediately but holds them for a period of time. In this waiting time, data packets arrive at the data packet queue for the node. The aggregation rate can be improved by the method, but this method increases the delay. The longer the waiting time is, the higher the data aggregation rate is. However, long waiting time causes an increase in the delay [44]. This method is only tested for networks with a star topology, and simulated tests corroborate the theoretical claims. The application scope is confined to star networks. Therefore, we explore an optimized data aggregation method [45,46] that adapts to the current practical application with low energy consumption and a small delay.

## 2.2. Research on Delay Optimization

Delay in WSNs refers to the difference between the time when the sensor nodes' data packets are generated and the time when the data packets are received by the sink. Emergencies and events in industrial production are monitored by WSNs. There are exacting requirements for the delay in WSNs, especially the WSNs for real-time monitoring. A data transmission delay could lead to catastrophic consequences. In the battlefield, fire monitoring, and factory automation control, the monitoring information is guaranteed to transfer quickly to the control center if an emergency occurs; otherwise, significant personal and property losses can be caused in the event of an enemy invasion, fire, or factory crash. Therefore, various methods to reduce delay are proposed.

### 2.2.1. Methods for Reducing Delay in a Wireless Unreliable Transmission Environment

Due to manufacturing costs and energy limits, the structure of sensor nodes is simple, and communication power is limited. Therefore, the wireless communication between nodes is often not reliable, sometimes as low as 60–90%. In this circumstance, methods with high data transmission reliability are used. These methods are divided into many categories:

(a) Retransmission mechanism. With the retransmission mechanism, the sender retransmits the data packets which are confirmed as lost [44]. The reliability of data transmission is improved by sending data many times. The first implementation of the retransmission mechanism is the send-wait retransmission mechanism. In this mechanism, the sender waits for the receiver to return an ACK that represents the packet sent by the sender has been received. If the sender successfully receives the ACK, this means that the receiver has received the data packet and the sender is able to send the next data packet. If the sender sends a packet and does not receive an ACK after a period of time, the packet is considered lost and is retransmitted. This phenomenon is called timeout retransmission. The process of sending a packet continues until the sender receives the receiver's ACK, or the number of retransmissions is greater than the predetermined threshold. After this step, the next packet is sent. The retransmission mechanism has an impact on the delay during the process of data transmission. It takes a long time before retransmission occurs, so delay increases due to repeated retransmissions. Networks with low transmission reliability experience great delays since a variety of retransmissions occur. To improve the defect of delay in the retransmission mechanism, several improved retransmission mechanisms are proposed. For example, the sender sends  $n$  packets at one time, and the receiver returns the sequence number of the received packets to the sender.

Then, the sender retransmits the packets with sequence numbers that are not returned. We present a retransmission mechanism in [44] in which delay is reduced. If the ACK is lost, superfluous data packets are transmitted by the sender. Compared to the common data packet, the ACK load is small. We propose that the receiver returns multiple same ACKs when it receives a data packet. Thus, the probability that the sender receives the ACK increases, the number of unnecessary retransmissions and the energy consumption decrease. Additionally, the total delay decreases since the number of retransmissions decreases.

(b) Data encoding mechanism. In the retransmission mechanism, the delay is long because of repeated retransmission. The main character of the data encoding mechanism is redundancy encoding [47]. Data can be recovered using redundant code even if the data are partially lost or incorrect. The drawback to this method is that the data packets are longer than the common data packets. The higher the redundancy, the higher the probability that the receiver recovers the data packet. However, the more redundant data the node sends, the more energy that is consumed [47–49].

### 2.2.2. The Delay in Duty Cycle Based Wireless Sensor Network

In many studies, nodes are always in a awake state, but sensor nodes are small, low cost, and battery powered. Their battery capacity is limited. The energy consumption of nodes in a wake state is 100–1000 times higher than in a sleep state. To save energy, we should try to keep nodes in a sleep state as much as possible. However, nodes stop sensing and communicating during the sleep state. One method used to save energy adopts the duty cycle mechanism [35]. Nodes periodically switch between the sleep state and the wake state. The ratio between the length of time of the wake state and the length of time of the whole cycle is called the duty cycle. The smaller the duty cycle, the more energy the nodes save. However, the disadvantage of this method is that delay increases. This is because the sender pauses sending data when the receiver is in a sleep state. There are several approaches to reduce delay in the duty cycle network. The basic idea of these methods is a tradeoff between network lifetime and delay. In [50], we proposed a method to reduce the delay in the dynamic duty cycle. There is residual energy in nodes which are far away from the sink in the WSNs. Increasing the duty cycle of the nodes reduces delay and does not affect the network lifetime.

### 2.2.3. Research on Delay Optimization for Adjusting Node Transmission Power

All of the above studies are based on unchanged transmission power. The transmission power of sensor nodes is adjustable; therefore, we use this characteristic to reduce delay. Because the receiving rate of the data packets is related to the transmission power of the sender, it is a useful method for increasing the transmission power of the node. High transmission power indicates a high signal to noise ratio (SNR). Thus, the success rate of receiving data packets increases, and the number of retransmissions and delay decreases. However, the lifetime of the network is affected by the transmission power. Therefore, this relationship between transmitting the power and delay reduction [2,26,51] should be studied.

### 2.2.4. Research on Delay in Data Aggregation

Studies on delay in data aggregation address that any number of data packets that are aggregated into one packet [42,43]. In this circumstance, only one packet is generated in a data collection cycle. A node aggregates all the data packets produced by its child node into one packet and sends it. In a cycle, nodes in the network send data in one time slot, while in other time slots, the network is monitored for receiving data. TDMA is widely used in these networks, and delay is the minimum number of slots needed in a data collection cycle. One time slot is the time it takes a node to send or receive a data packet. There are many studies in this area. For example, Huang et al. [45] proposed an algorithm with an upper bound of  $23R + \Delta - 18$  time slots in delay of aggregation, where  $R$  is the network radius and  $\Delta$  is the maximum node degree. Xu et al. [46] proposed an algorithm and proved the delay of the aggregation schedule generated by their algorithm was at most,  $16R + \Delta - 14$  time slots.



We also proposed a protocol named the broadcasting combined with multi-NACK/ACK (BCMNA) protocol which combines the energy efficiency and minimizes delay under a statistically reliable constraint in [41]. Previous research is based on reliable wireless communications, but packets are often lost in wireless communication. The objective of the BCMNA protocol is to minimize the delay in unreliable wireless communication during the data collection process. In the BCMNA protocol, the network is optimized using the following methods. Data collection is divided into two parts, intra-cluster data collection, and inter-clusters data collection. After intra-cluster data collection, the cluster head nodes broadcast NACK to inform the nodes which fail to send data to resend the data packets. The method of sending multiple NACKs is adopted to reduce the energy consumption of the cluster head node and the total delay by decreasing the frequency of ACK transmission. In the inter-cluster data collection, multiple same ACKs are returned when a data packet is received by the receiver. Although the number of ACKs increases, the number of retransmitting data packets and the energy consumption of the cluster head nodes decrease. As a result, the lifetime of the network increases, and the total delay decreases.

In the clustering network, finding an energy-efficient policy to opt cluster heads (CHs) in the WSNs has become increasingly important. This importance is closely related to such factors as the network lifetime and efficiency. LEACH [52] is a classic cluster head selection algorithm, but it does not consider the nodes' heterogeneities. Accordingly, SEP is proposed for clustered heterogeneous wireless sensor networks. Based on SEP, prolong-SEP [53] etc. are designed to increase the lifetime. In the field of ad hoc networks, dynamic Doppler velocity clustering (DDVC) [54] and the hierarchical clustering algorithm (HCAL) [55] are proposed to improve clustering stability and performance of the network.

### 2.2.5. Other Research Related to Delay

Some strategies, though not designed to reduce delay, also help reduce delay. For example, the multipath routing strategy [7,32]. This routing strategy was proposed for network attacks. To prevent the attacker from dropping packets, the data is transmitted on multiple routing paths from the source to the sink. The time difference from when the first data packet reaches the sink is the delay of data transmission. This research is related to the research in this paper. The current WSNs have been rapidly developed in combination with edge networks [56,57], IoT [58] and cloud networks [59–63], which are the main networks for data collection [64].

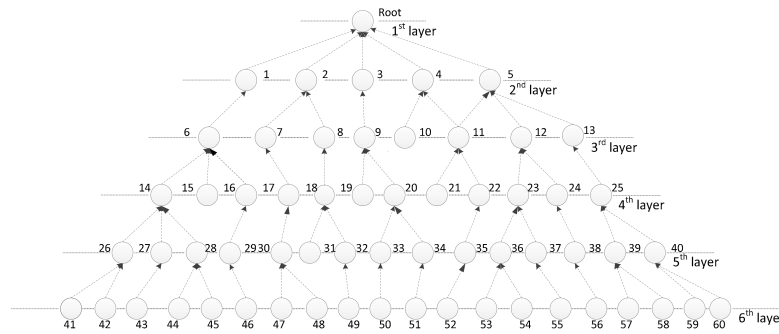
## 3. System Model and Problem Statement

### 3.1. System Model

The application scope of the AAR scheme is the planar WSNs. Similar to most research, the planar WSNs can be abstracted as tree networks, in which the sink is the root of the tree network [39,40], but it's not a node in the network. All of the data packets produced by these sensors are aggregated and transmitted to the sink. A model contains  $m$  layers, and each layer has  $n_i$  nodes. This is shown below in Equation (1):

$$\Sigma = \begin{bmatrix} \sigma_2^1 & \sigma_2^2 & \cdots & \sigma_2^{n_2} \\ \sigma_3^1 & \sigma_3^2 & \cdots & \sigma_3^{n_3} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_m^1 & \sigma_m^2 & \cdots & \sigma_m^{n_m} \end{bmatrix} \quad (1)$$

Each node has its fixed level and a relative position, and there is only one default parent node. This tree-shaped wireless network is illustrated in Figure 1. As shown in Figure 1, the network has six layers and 60 nodes. Each node has a default parent node. For example, node 34 is the parent node of node 51.



**Figure 1.** Instance of the system model.

The activities of the nodes in the network are periodic. In the data collection process  $\mathcal{U}$ , the cycle  $u$  is divided into the packet generation period  $v_1$  and the aggregation and transmission period  $v_2$ :

$$\mathcal{U} = \{u_1, u_2, \dots, u_n\}, u = \{v_1, v_2\} \quad (2)$$

Each node in the data packet generation period  $v_1$  receives the data aggregation queues sent by their child nodes, and at most, one data packet is produced by the nodes. Each node in the aggregation and transmission period determines whether to aggregate and transmit. The transmission of the aggregation queue is performed on two adjacent layers. A node that sends the aggregation queue is  $\sigma_i^j$ , and a node which receives the queue is considered to  $\sigma_{i-1}^k$  during an aggregation and transmission period. We set a unit time as a cycle  $u$ . The cycle of all nodes is synchronous. In this system, the nodes in the network are homogenous.

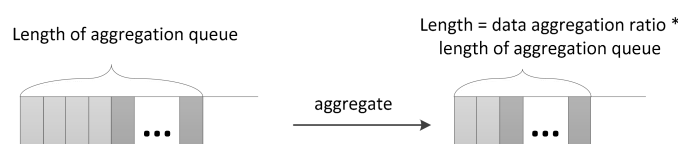
*Definition 1 (probability of generating a packet)*

The attribute of applications is modeled using statistical analysis [29]. For example, a wearable device generates packets for a period of 30 percent of the entire cycle, and no packet is generated during the rest period. Thus, the traffic is modeled as a successive Bernoulli trial based on a discrete time index  $t$  with a certain probability  $P_\alpha$  [65]. Probability of generating a packet is defined as the probability that a node generates a packet in a cycle. In a packet generation period, the packets of all nodes in the model are generated as shown in Equation (3):

$$C = \begin{bmatrix} c_2^1 & c_2^2 & \cdots & c_2^{n_1} \\ c_3^1 & c_3^2 & \cdots & c_3^{n_2} \\ \vdots & \vdots & \ddots & \vdots \\ c_m^1 & c_m^2 & \cdots & c_m^{n_m} \end{bmatrix} \quad c \in \{0, 1\} \quad (3)$$

*Definition 2 (data aggregation ratio)*

After data aggregation, the redundant data is removed, and the packet queue length  $l_i^j$  sent by the nodes in the lower layer is smaller than the original queue length. The ratio of the length of the queue after data aggregation to the length before data aggregation is defined as the data aggregation ratio  $\lambda$  (see from Figure 2).



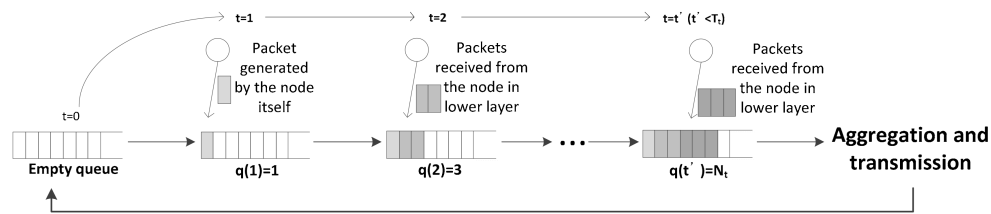
**Figure 2.** Length of the aggregation queue is reduced by aggregation.

The packets queue transmitted by the nodes is denoted as Equation (4):

$$Q = \begin{bmatrix} q_2^1 & q_2^2 & \cdots & q_2^{n_2} \\ q_3^1 & q_3^2 & \cdots & q_3^{n_3} \\ \vdots & \vdots & \ddots & \vdots \\ q_m^1 & q_m^2 & \cdots & q_m^{n_m} \end{bmatrix} \quad q = \begin{cases} \lambda \cdot l(\sigma_i^j) \\ 0 \end{cases} \quad (4)$$

**Definition 3** (packet aggregation threshold)

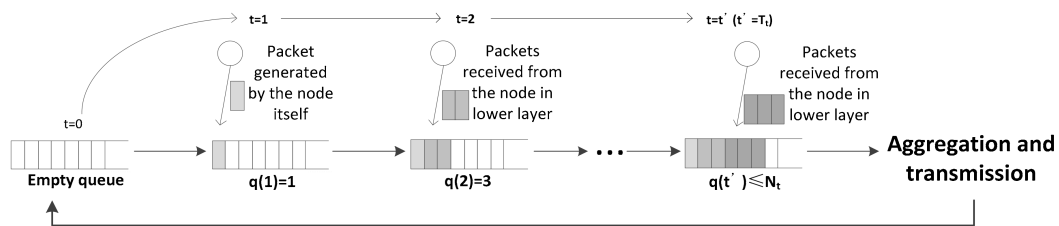
The nodes in the model store the incoming data packets in the aggregation queue. If the number of queued packets is the same as or greater than the predetermined packet aggregation threshold, the queued packets are transmitted as an aggregated packet queue in the aggregation and transmission period (see from Figure 3). The aggregation threshold refers to the maximum number of packets aggregated into a single frame. We denote this predetermined packet aggregation threshold as  $N_t$ .



**Figure 3.** Packet aggregation occurs since the number of packets is equal to or greater than  $N_t$ .

**Definition 4.** (Value of the packet aggregation timer)

Because of the characteristics of the wireless network, variable waiting (or queuing) delay occurs during the aggregation process. If the time taken for the number of queued packets is the same as, or greater than packet aggregation the threshold is excessively long, the aggregation delay at the aggregator significantly increases, and the QoS deteriorates [29]. To prevent this excessive delay, the node transmits an aggregated frame when the aggregation timer expires, although the number of packets in a queue is less than the packet aggregation threshold (see from Figure 4). The aggregation timer refers to the maximum allowable time to wait for packets before transmission. We denote the value of the packet aggregation timer as  $T_t$ .



**Figure 4.** Packet aggregation occurs since the value of the aggregation timer is equal to  $T_t$ .

### 3.2. Problem Statement

- (1) Maximize the efficiency of data packet aggregation.

In the network, increasing the aggregation efficiency is an effective way to improve the QoS and reduce the delay and the energy consumption.  $\varphi$  is considered the upper limit of the number of aggregations, and the number of aggregations performed in a node in the network is  $\gamma$ . The formula to maximize the data packet aggregation efficiency  $\Xi$  can be expressed by the following Equation (5):



$$\text{Max}(\Xi) = \min \frac{\wp}{\sum_{i=2}^m \sum_{j=2}^{n_i} \gamma_i^j} \quad (5)$$

(2) Minimize the delay during the data collection process.

This optimization method aims to reduce the total network transmission delay  $\mathcal{D}$ . The total delay in the AAR scheme is the sum of delay  $\mathcal{D}_i^j$ , which is shown as the following Equation (6):

$$\text{Min}(\mathcal{D}) = \text{Min} \left( \sum_{i=2}^m \sum_{j=2}^{n_i} \mathcal{D}_i^j \right) \quad (6)$$

In the AAR scheme, the delay of each node is reduced. The delay in the common scheme is considered as  $\mathcal{D}_i^j$ . It can be described by Equation (7):

$$\mathcal{D}_i^j \leq \mathcal{D}_i^j | \forall i \in [2, m], j \in [2, n_i] \quad (7)$$

Due to the requirement of low power consumption of the wireless network, the number of data aggregations cannot be increased excessively while optimizing the delay.

(3) Maximize network lifetime

Network lifetime is defined as the death time of the first node in the network [32,35,36]. Considering that the average energy consumption of the  $i$ -th node in the network is  $\omega_i$ , its initial energy is  $E_{ini}^i$ , and there are  $\mathcal{N}$  nodes in the network. To maximize the lifetime of the network, the network lifetime of the first node to die in the network should be maximized. Therefore, Equation (8) can be obtained:

$$\max(\mathcal{L}) = \text{Max} \left[ \min_{1 \leq i \leq \mathcal{N}} \left( E_{ini}^i / \omega_i \right) \right] \quad (8)$$

In summary, the objectives of this research are as follows in Equation (9):

$$\left\{ \begin{array}{l} \text{Max}(\Xi) = \min \frac{\wp}{\sum_{i=2}^m \sum_{j=2}^{n_i} \gamma_i^j} \\ \text{Min}(\mathcal{D}) = \text{Min} \left( \sum_{i=2}^m \sum_{j=2}^{n_i} \mathcal{D}_i^j \right) \\ \max(\mathcal{L}) = \text{Max} \left[ \min_{1 \leq i \leq \mathcal{N}} \left( E_{ini}^i / \omega_i \right) \right] \end{array} \right. \quad (9)$$

#### 4. Optimization Mechanism Design

To clearly state the parameters of this paper, the parameters introduced in this paper can be found in Table 1.

We propose the AAR scheme to reduce the total delay in wireless networks. A detailed description of the AAR scheme is given below. In the process of data transmission in two adjacent layers, the scheme first selects the nodes with a long data queue. Next, these nodes' priorities are set by the length of the data queue. The aggregation queues are assigned to these nodes by the priority. In the process of queue assignment, the AAR scheme ensures that the nodes receive sufficient data packets and perform data aggregation. As a result, the frequency of aggregation increases and the delay is significantly reduced.

In the common scheme, each node only receives data packets from the fixed child node and sends the aggregation queue to the fixed parent node. It is possible that the waiting queue in the node is long but it cannot receive sufficient data packets to perform aggregation, thus delay increases excessively. The AAR scheme solves this problem.

To achieve the AAR scheme, we propose a node assignment algorithm (NAAL), which is based on the greedy strategy. We describe and explain this algorithm in Section 4.1, then analyze the complexity of NAAL in Section 4.2. Finally, we provide an instance to demonstrate this algorithm.

**Table 1.** Parameter description.

Parameter	State
$L_i$	All the nodes in layer $i$ of the wireless network.
$\Sigma_i$	The set which contains the nodes which participate in the transmission in layer $i$ .
$\sigma_i^j$	A node at $L_i$ and relative position $j$ .
$\mathcal{N}$	The number of the nodes in the network.
$m$	The number of the layer in the network.
$n_i$	The number of the nodes at $L_i$ .
$N_t$	Packet aggregation threshold.
$T_t$	Value of the packet aggregation timer.
$\tau(\sigma_i^j)$	The parent node of $\sigma_i^j$ by default.
$l(\sigma_i^j)$	Current length of data packets queue of $\sigma_i^j$ .
$t(\sigma_i^j)$	Current value of aggregation timer of $\sigma_i^j$ .
$\omega_i^j$	The ratio of the length of the current queue to $N_t$ in $\sigma_i^j$ .
$\Omega$	The parameter at which a node receives the aggregate data queue from the lower layer.
$\lambda$	Data aggregation ratio.
$P_a$	The probability that a sensor generates sensing data during a packet generation period.

#### 4.1. Description and Remarks about NAAL

The aim of NAAL is to assign the aggregation data queues to the nodes with sufficiently long data waiting queues to make them aggregate data. We set a parameter  $\Omega$ . If the ratio of the length of the waiting queue to the aggregation threshold  $N_t$  is equal to or greater than  $\Omega$  and less than 1, the node receives the queue. The proposed algorithm is given below in Algorithm 1.

##### Algorithm 1 Node assignment algorithm

```

1: For each  $\sigma_i^j \mid j$  from 1 to  $n_i$  // collect all the nodes whose aggregation queue is ready to aggregate.
2:   If  $l(\sigma_i^j) \geq N_t$  or  $t(\sigma_i^j) \geq T_t$  then
3:      $\sigma_i^j \in \Sigma_i$ 
4:   End if
5: End for
6: If  $\Sigma_i = \emptyset$  then // if  $\Sigma_i = \emptyset$ , no node in  $L_i$  send the aggregation queue.
7:   Return
8: End if
9: For each  $\sigma_{i-1}^k \mid k$  from 1 to  $n_{i-1}$  // collect all the nodes whose  $\Omega \leq \omega_{i-1}^k < 1$ .
10:  If  $\Omega \leq \omega_{i-1}^k < 1$  then
11:     $\sigma_{i-1}^k \in \Sigma_{i-1}$ 
12:  End if
13: End for
14: For each  $\sigma_{i-1}^{k'}$   $\mid$  from  $\sigma_{i-1}^k$  with max  $\omega_{i-1}^k$  to  $\sigma_{i-1}^{k'}$  with min  $\omega_{i-1}^{k'}$  in  $\Sigma_{i-1}$  // assign  $\sigma_{i-1}^{k'}$  to  $\sigma_i^j$  with the
    priority of large  $\omega_{i-1}^k$  to small.
15:   While  $l(\sigma_{i-1}^{k'}) < N_t$  and  $\Sigma_i \neq \emptyset$  then // assign  $\sigma_{i-1}^{k'}$  with sufficient aggregation queues to make
    the node aggregate.
16:      $l(\sigma_{i-1}^{k'}) = l(\sigma_{i-1}^{k'}) + l(\sigma_i^j) * \lambda$ 
17:      $\sigma_i^j \notin \Sigma_i$ 
18:   End while
19:    $\sigma_{i-1}^{k'} \notin \Sigma_{i-1}$ 
20: End for
21: If  $\Sigma_i \neq \emptyset$  then // the remaining queues of nodes in  $\Sigma_i$  are transmitted to  $\tau(\sigma_i^j)$ .
22:   While  $\Sigma_i \neq \emptyset$  then
23:      $l(\tau(\sigma_{i-1}^{j''})) = l(\tau(\sigma_{i-1}^{j''})) + l(\sigma_i^{j''}) * \lambda$ 
24:      $\sigma_i^{j''} \notin \Sigma_i$ 
25:   End while
26: End if

```

The explanatory remarks about Algorithm 1 are shown below:

*Line 1–5:* When traversing a layer  $L_i$ , all the nodes in this layer that meet the aggregation criteria are moved to  $\Sigma_i$ .

*Line 6–8:* If  $\Sigma_i$  is empty, the algorithm would return.

*Line 11–13:* select the nodes in  $L_{i-1}$  whose  $\omega_{i-1}^k$  is greater than or equal to  $\Omega$  and less than 1 to  $\Sigma_{i-1}$ .

*Line 14–20:* Nodes in  $\Sigma_{i-1}$  are prioritized with  $\omega$  from large to small. Next, assign the aggregation queue of node in  $\Sigma_i$  to  $\sigma_{i-1}^{k'}$  by priority and remove the node from  $\Sigma_i$  until all the aggregation queues provided by nodes in  $\Sigma_i$  makes  $l(\sigma_{i-1}^{k'}) \geq N_t$ , then remove  $\sigma_{i-1}^{k'}$  from  $\Sigma_{i-1}$ .

*Line 21–26:* If there are nodes remaining in  $\Sigma_{i-1}$  which have no queue to receive, the algorithm ends. If  $\Sigma_i$  is not empty, transmit each aggregation queue to its parent node by default  $\tau(\sigma_i^{j''})$ .

#### 4.2. NAAL Complexity Analysis

We define the number of nodes in the network as  $\mathcal{N}$ , the number of nodes in layer  $i$  is  $N_i$ . The number of layers in the network is  $m$ . First, the nodes in layer  $i$  which perform aggregation are moved to  $\Sigma_i$ . The overhead of this step is  $O(N_i)$ . Then, we move the nodes with a long data queue in layer  $i-1$  to  $\Sigma_{i-1}$ . The time complexity is  $O(N_{i-1})$ . After that, the nodes in  $\Sigma_{i-1}$  are sorted by the length of the data queue. The time complexity is  $O(\log N_{i-1})$ . Finally, assign the data queues to the nodes in  $\Sigma_{i-1}$  and handle with the remaining nodes in  $\Sigma_{i-1}$  or  $\Sigma_i$ . It performs the nodes traversal in  $\Sigma_i$  and  $\Sigma_{i-1}$ . The time complexity is  $O(N_i + N_{i-1})$ . Therefore, the time complexity of assigning the data queues in the whole network one time using the proposed algorithm is shown as Equation (10):

$$O\left(2 \sum_{i=2}^{i=m-1} N_i + 2 \sum_{j=3}^{j=m} N_j + \log \sum_{i=2}^{i=m-1} N_i\right) = O(\mathcal{N}) \quad (10)$$

#### 4.3. Illustration of NAAL

The detailed process of the algorithm is clarified in this sub-section. As revealed in Figure 5, the nodes of the two layers are described,  $\Sigma_{i-1}$  is {a, c, f}, and  $\Omega \leq l(\sigma_{i-1}^c) \leq l(\sigma_{i-1}^f) \leq l(\sigma_{i-1}^a) \leq 1$ . Node  $i$  is assigned to node  $a$  and then  $i$  is removed from  $\Sigma_i$ . At this time, the queue of node  $a$  can be aggregated, and  $a$  is removed from  $\Sigma_{i-1}$ . Next, assign node  $m$  to  $f$  and remove  $m$  from  $\Sigma_i$ . The sum of the length of the queue in  $f$  and the length of the queue received from  $m$  is greater than  $N_t$ , exclude  $m$  from  $\Sigma_{i-1}$ . The queue of  $n$  is assigned to  $c$  and  $n$  is excluded from  $\Sigma_i$ . However,  $c$  must obtain additional data to perform aggregation. Thus, the data queue of  $r$  is assigned to be transmitted to  $c$ . The queue meets the criteria of aggregation, exclude  $c$  from  $\Sigma_{i-1}$ . Now,  $\Sigma_{i-1}$  is an empty set, but node  $s$  is still in  $\Sigma_i$ . Therefore,  $s$  would transmit the queue to its default parent node  $g$ .

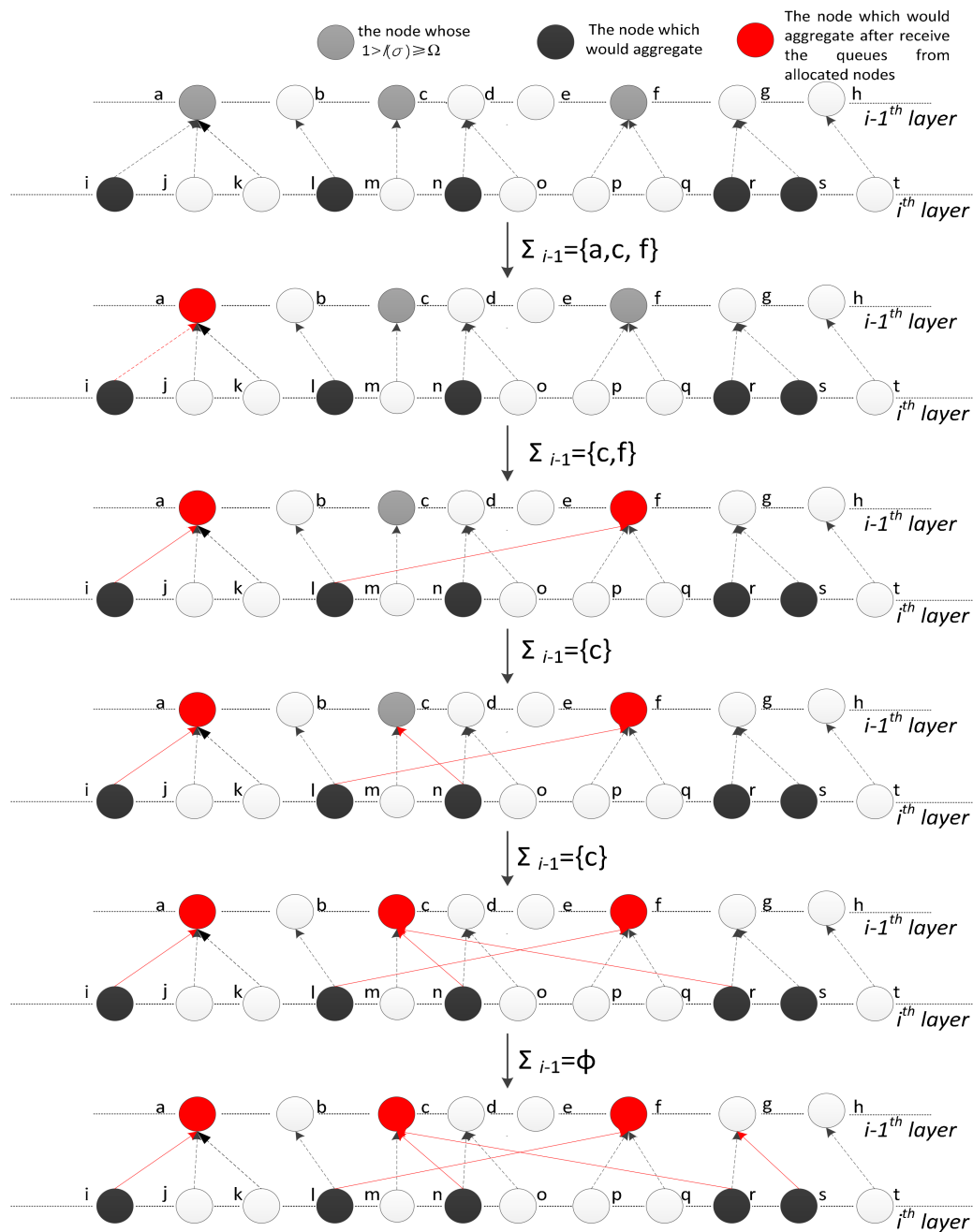


Figure 5. Demonstration diagram of the algorithm.

## 5. Performance Analysis and Optimization

### 5.1. Methodology and Setup

In this section, we compare the AAR scheme with the common scheme (CS) regarding total delay, the number of aggregations and the lifetime. The main characteristic of CS is that each node's aggregation queue can only be sent to the default parent node. We performed Monte Carlo simulations to validate the optimization performance of our method. First, the simulation process generates a tree network randomly. Each node has a fixed parent node, which is used to receive the queues sent by the node. Fifty thousand packets are generated in the network, and when all these packets are received by the root, it is regarded as the end of a simulation. We set a cycle as a unit time that contains one period of packet generation and one period of aggregation.

The network parameters used in our experiments are listed in Table 2. The simulation networks include different numbers of layers and nodes.  $m = 3$  indicates that the number of layers in the network is 3. To evaluate the performance of the AAR scheme in different environments, we build three scenarios.

Table 2. Simulation parameters.

Parameter	Value
$m$	{3, 5, 7}
$\mathcal{N}$	{20, 60, 80}
$\lambda$	{0.2, 0.5, 0.8}
$P_a$	{0.3, 0.6, 0.9}
$N_t$	5–50 (packet)
$T_t$	5–50 (unit)
$\Omega$	{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0}

The parameters which determine the scale of the network are  $m$  and  $\mathcal{N}$ .  $\mathcal{N}/m$  is used as the indicator that measures the structure of the tree. A large value of  $\mathcal{N}/m$  means that this is a fat network. The reason for using this indicator is as follows: we set three kinds of network,  $(m, \mathcal{N}) = \{(7, 80), (7, 120), (5, 80)\}$ . The value of  $\mathcal{N}/m$  is 1.33, 1.5, 1.5 (the sink is in the first layer, but it is not a node in the network). As shown in Figure 6, the networks with the same  $\mathcal{N}/m$  have the same value of average delay.

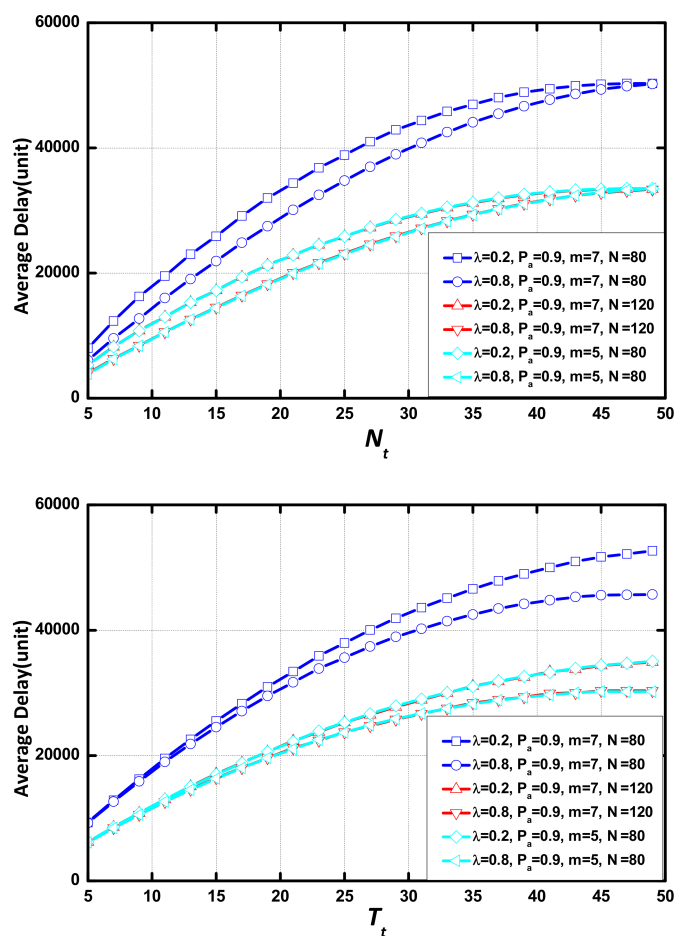


Figure 6. Average delay in various circumstance vs.  $N_t$  or  $T_t$ .

The scenarios in the simulation are:  $(m, \mathcal{N}) = \{(3, 20), (5, 60), (7, 80)\}$ . The values of  $\mathcal{N}/m$  of the three scenarios are 1.0, 1.5, 1.33 respectively.  $\lambda$ ,  $P_a$  are considered as the environmental parameters in the simulation. The values of  $\lambda$  and  $P_a$  are selected randomly.

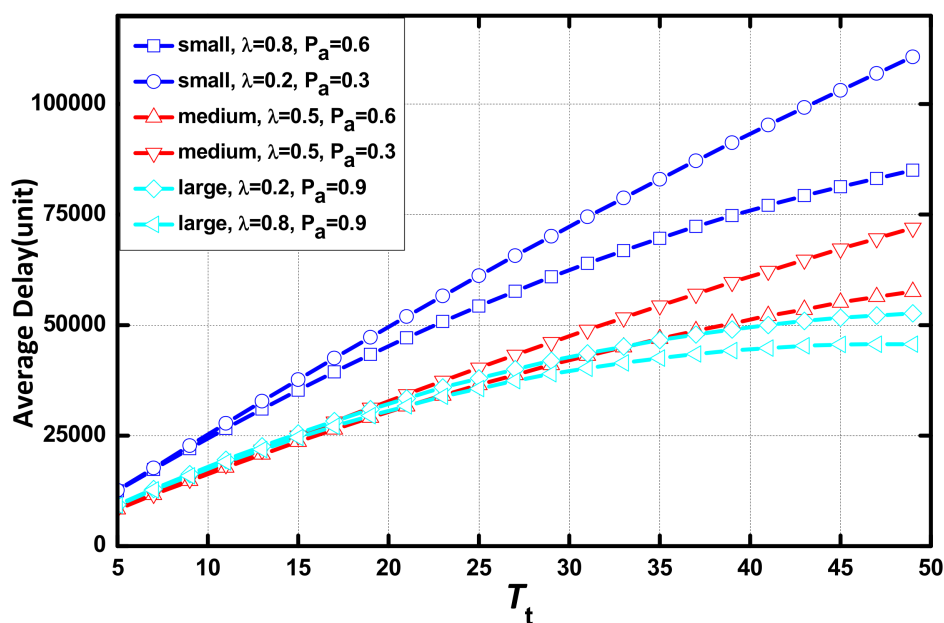
We first study the performance of the optimization in various cases in terms of delay in Section 5.2. In Section 5.3, we study the effect of  $\Omega$  on the optimization performance. Finally, we compare the performance of the AAR scheme and the improved AAR scheme in Section 5.4. All simulations using the same settings are repeated twenty times to get the average values.

## 5.2. Optimization Performances on Delay

We aim to investigate the optimization effect of the AAR scheme in different circumstances. We analyze the optimization performance of the AAR scheme in different scenarios in terms of delay. The effect of environmental parameters,  $\lambda$  and  $P_a$ , on delay and the number of aggregation is tested. Next, the optimization performance of delay and lifetime and the effect on the number of aggregation are presented. The lifetime of the networks using these two schemes are compared. In this sub-section,  $\Omega$  is a constant with the value of 0.4.

### 5.2.1. The Effect of Environmental Parameters on Delay

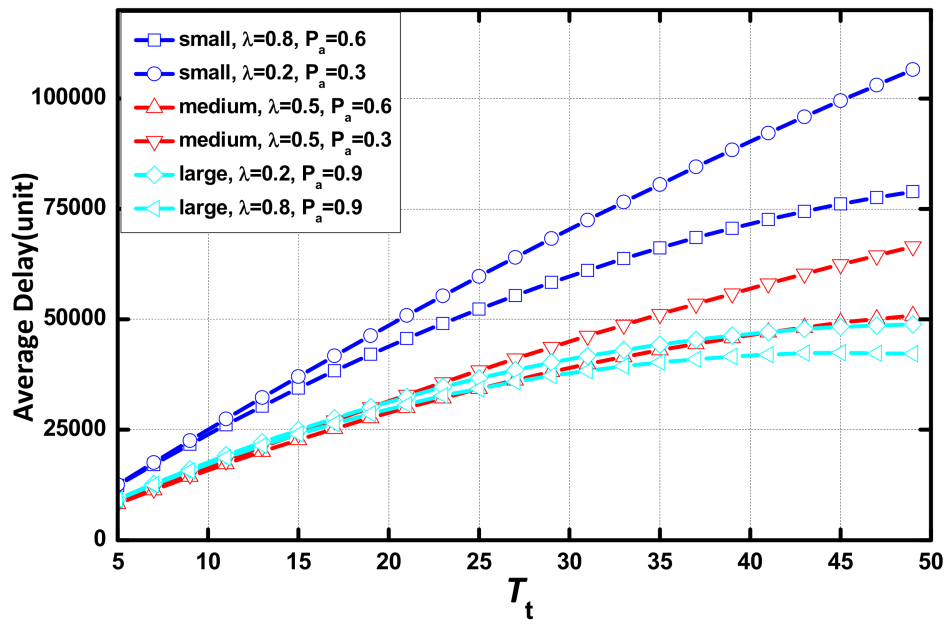
As shown in Figure 7, delay increases linearly as  $T_t$  increases. When  $T_t$  is small, there is no difference in the delay between networks in different environments in each scenario and the difference increases with the increase of  $T_t$ . In the same scenario, the network with a large value of  $\lambda$  or  $P_a$  presents a small delay. The increasing rate of delay is lower, and increasing delay disappears faster when the value of  $\lambda$  or  $P_a$  increases. In addition, the AAR scheme causes no change to the effect of  $T_t$  on delay.



(a) Common scheme

Figure 7. Cont.





(b) AAR scheme

Figure 7. Average delay vs.  $T_t$  in different scenarios.

In Figure 8, we compare the delay of the networks in various environments in the same scenario. The values of the environmental parameters is:  $(\lambda, P_a) = \{(0.8, 0.3), (0.8, 0.9), (0.8, 0.6), (0.2, 0.6), (0.5, 0.6)\}$ . It can be seen that  $P_a$  is the environmental parameter that determines the delay, and  $\lambda$  has little effect on delay. Delay is affected significantly by the change of  $P_a$  when the value of  $P_a$  is small.

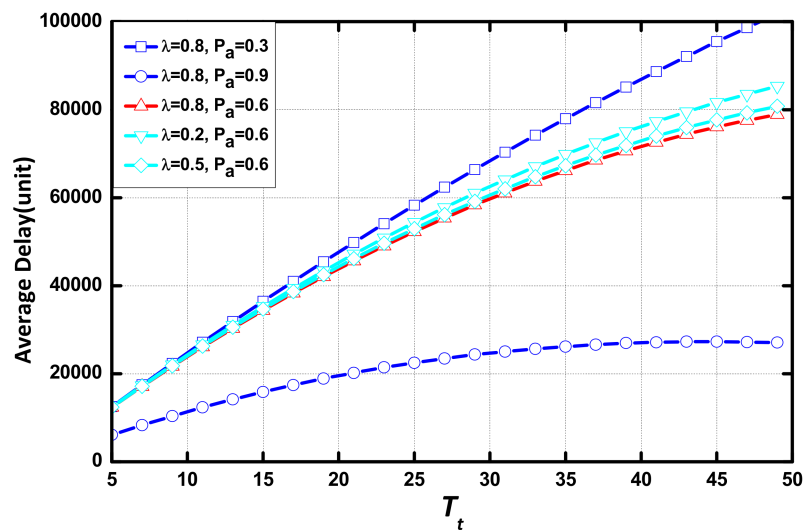
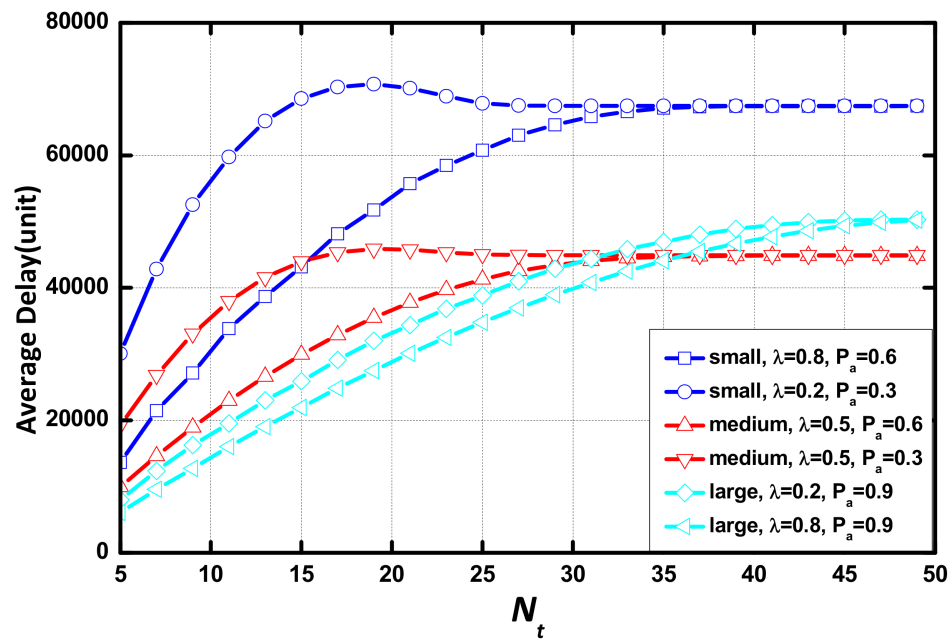
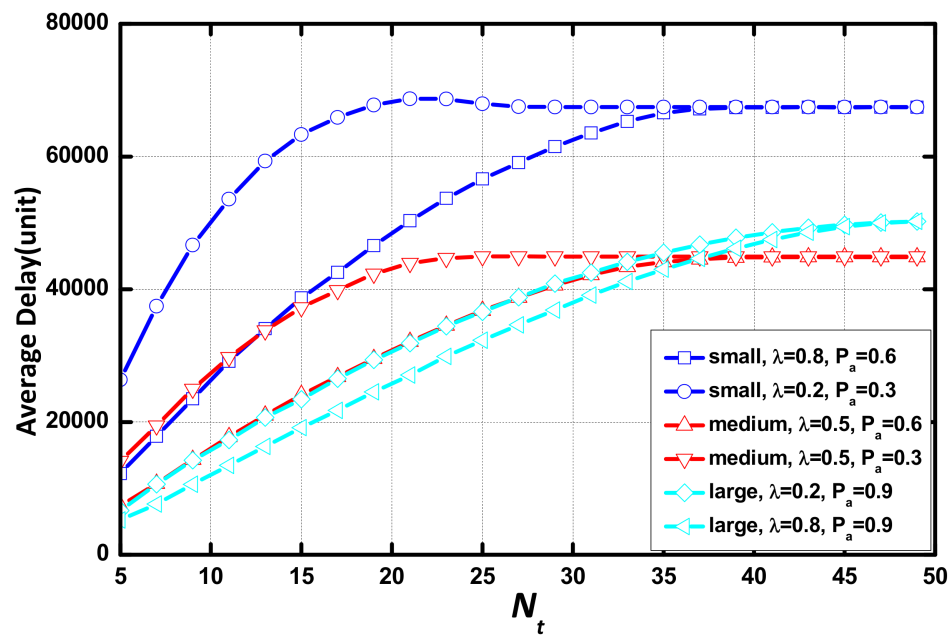
Figure 8. Average delay vs.  $T_t$  with different  $(\lambda, P_a)$ .

Figure 9 indicate that the delay significantly increases and then remains unchanged as  $T_t$  increases. The difference in delay between the network with different environmental parameters in each scenario increase at first and then decrease to 0. The unchanged delay points out that  $N_t$  is not a parameter that determines delay in these cases.  $T_t$  is the only parameter to affect delay. In this case, all aggregations occur because of the expiration of aggregation timer. The AAR scheme makes causes no change to the effect of  $N_t$  on delay.



(a) Common scheme



(b) AAR scheme

**Figure 9.** Average delay vs.  $N_t$  in different scenarios.

Figure 10 compares the delay of the networks in various environments. It can be seen that  $P_a$  is the major environmental parameter. However, delay is saturated at a large aggregation threshold eventually.

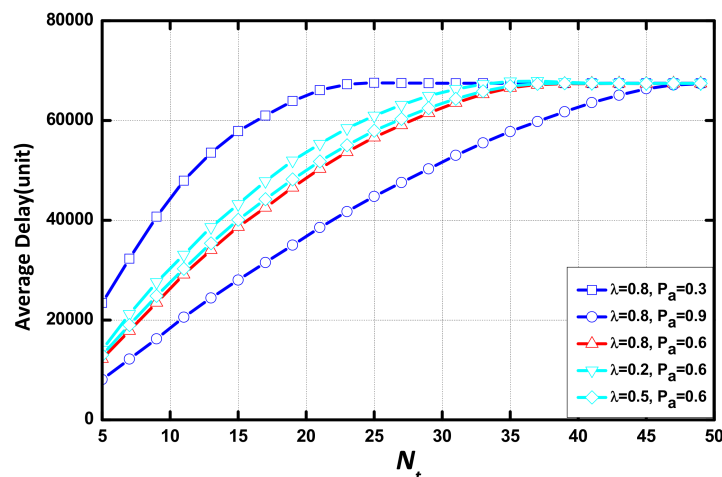


Figure 10. Average delay vs.  $N_t$  with different  $(\lambda, P_a)$  when  $(m, N) = (3, 20)$ .

### 5.2.2. The Effect of Environmental Parameters on the Number of Aggregations

In this subsection, we study the effect of environmental parameters regarding the number of aggregations. As revealed in Figure 11, the number of aggregations rapidly decreases at first but is saturated at a sufficiently large value of  $N_t$  and  $T_t$ . By the contrast of (a), (b) and (c), (d), with the increase of  $T_t$ , the number of aggregations decreases more than with the increase of  $N_t$ . When the value of  $T_t$  is large, the difference of the number of aggregations in these networks is smaller than when the value of  $N_t$  is large. In each scenario, the average number of aggregation decreases with the small  $\lambda$  or the large  $P_a$ . The number of packets transmitted in the simulations is constant. The small number of aggregations represents a high expectation of the number of data packets per aggregation. The number of aggregations decreases smoothly at a large value of  $\lambda$  or  $P_a$ . The trend of the number of aggregations with  $N_t$  or  $T_t$  is opposite to that of delay, and the area that the number of aggregation doesn't change is similar to that of delay. Additionally, the AAR scheme has no effect on the trend of the number of aggregations.

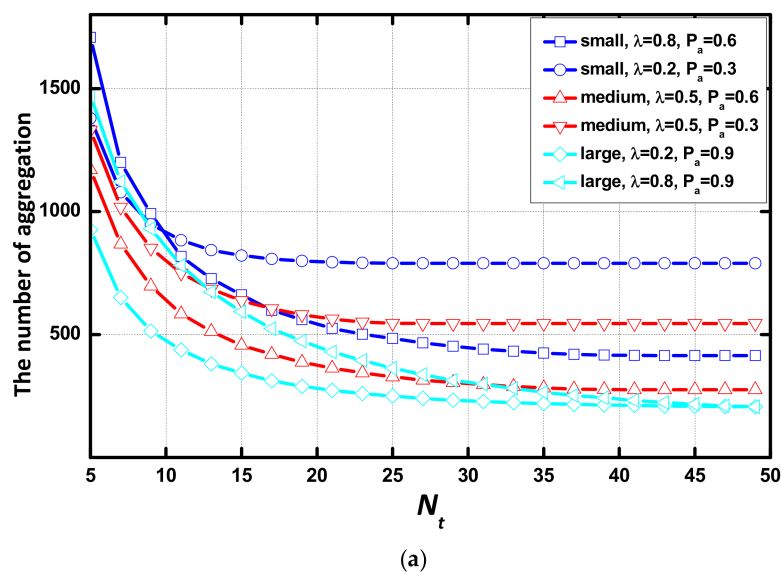
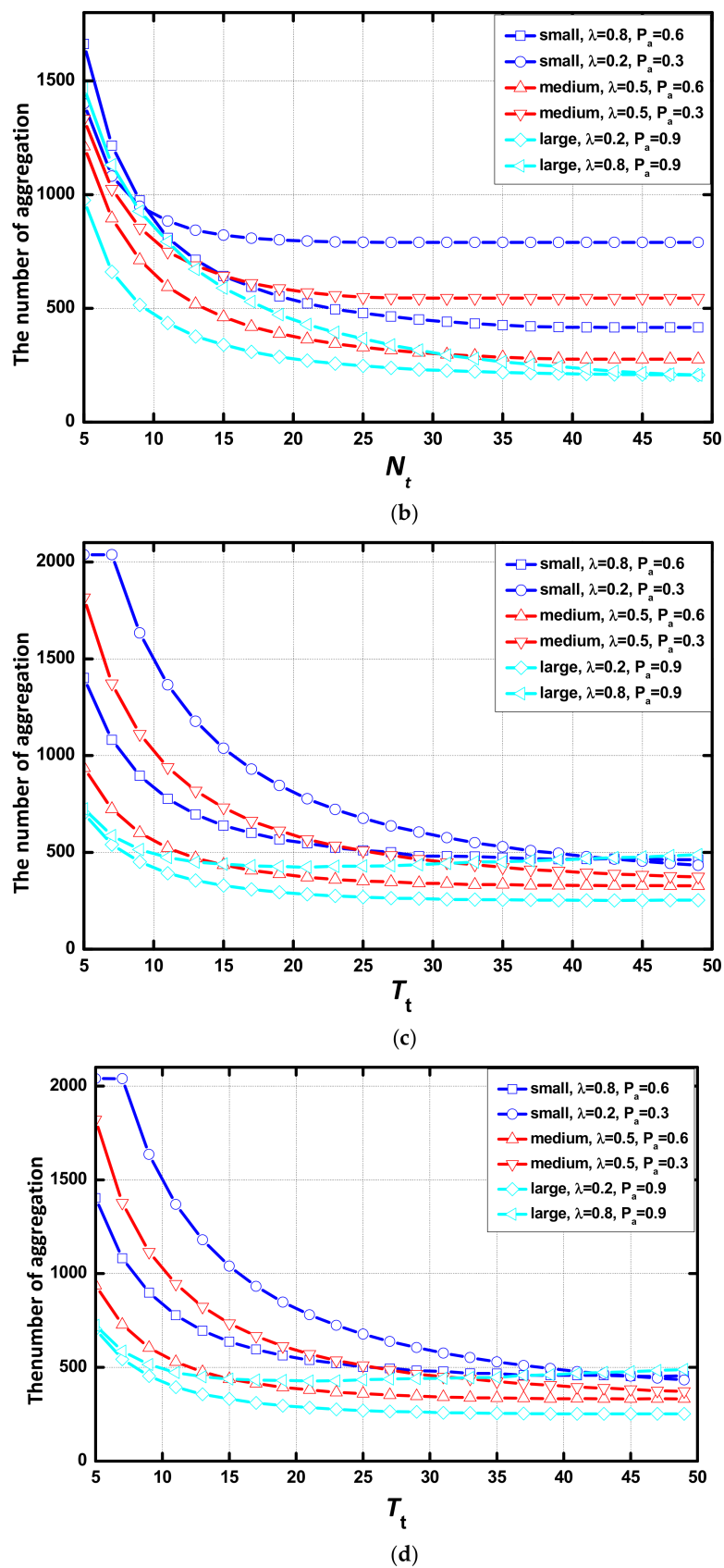
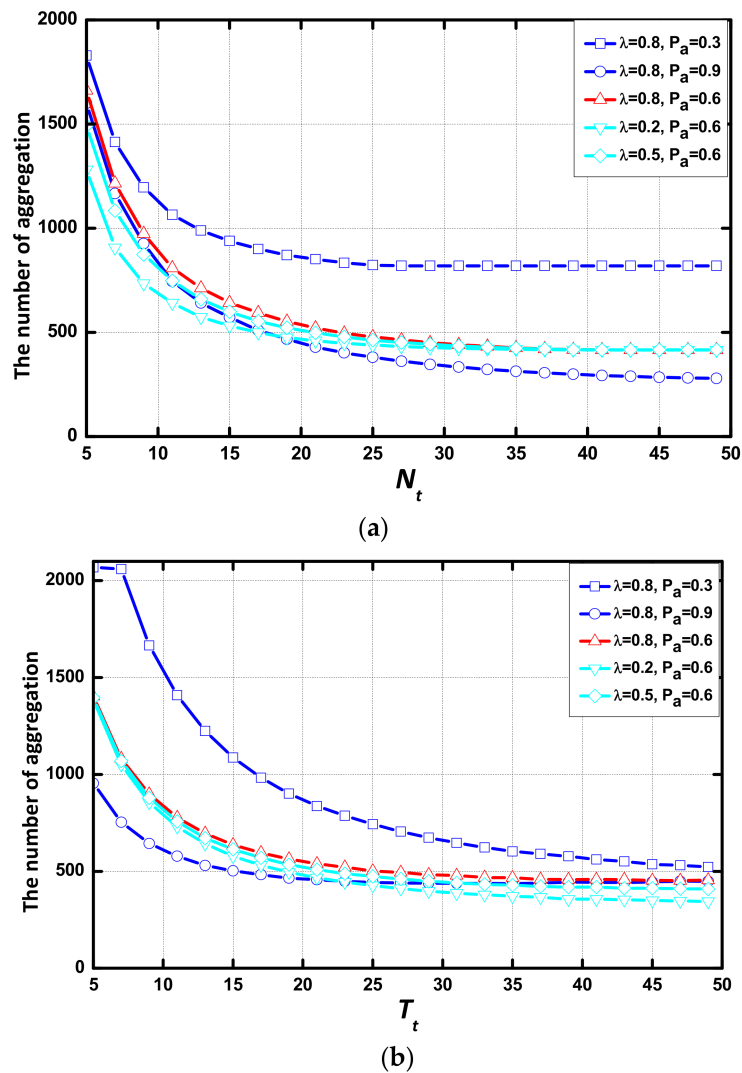


Figure 11. Cont.



**Figure 11.** Number of aggregations vs.  $N_t$  or  $T_t$  in different environments (a) Common scheme, vs.  $N_t$ ; (b) AAR scheme, vs.  $N_t$ ; (c) Common scheme, vs.  $T_t$ ; (d) AAR scheme, vs.  $T_t$ .

As seen in Figure 12, there is a gap in the number of aggregations between the networks with different  $P_a$ .  $P_a$  is the major environmental parameter which determines the number of aggregations.



**Figure 12.** Average delay with different  $(\lambda, P_a)$  when  $(m, N) = (3, 20)$  (a) Average delay vs.  $N_t$ ; (b) Average delay vs.  $T_t$ .

### 5.2.3. The Optimization Performance of AAR Scheme

As seen in Figure 13, the ratio between the delay with the AAR scheme and the delay with CS linearly decreases as  $T_t$  increases. The optimization performance of the AAR scheme increases as  $T_t$  increases. Additionally, there is no sign that the trend would stop with the increase of  $T_t$ . In the same scenario, the network with the larger value of  $\lambda$  or  $P_a$  has better optimization performance.

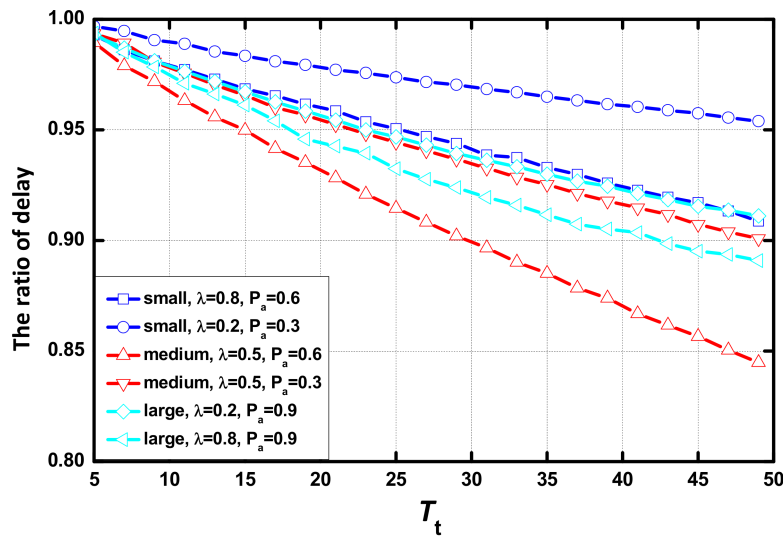


Figure 13. Ratio of average delay of the AAR scheme and the common scheme vs.  $T_t$ .

Figure 14 illustrates the ratio between the delay with the AAR scheme and the delay with the CS. The horizontal bar indicates the increase of  $N_t$ . With the increase of  $N_t$ , the ratio increases rapidly and is saturated at a large value of  $N_t$ . The value of  $N/m$  is positively correlated with the mathematical expectation of the aggregation queue that each layer provide. The value of  $\lambda$  and  $P_a$  determine the area that has optimized performance and the value of  $N/m$  determines the performance of AAR scheme. The area where the AAR scheme has no optimization is similar to the area where delay no longer increases with the increases of  $N_t$ . In summary, the AAR scheme works well when  $N_t$  is the parameter that determines the performance of the packet aggregation process. When  $N_t$  and  $T_t$  are constant, the area where the AAR scheme works is constrained by  $\lambda$  and  $P_a$  and the performance is positively correlated with the value of  $N/m$ .

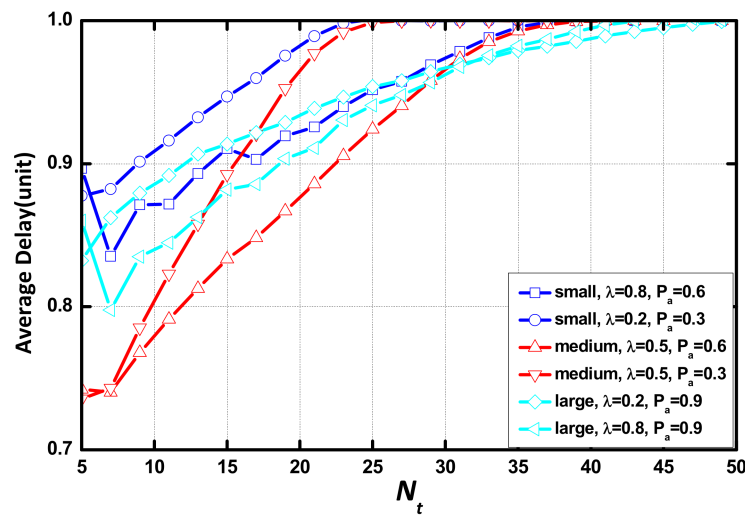


Figure 14. Ratio of the average delay of the AAR scheme and the common scheme vs.  $N_t$ .

Network lifetime is defined as the death time of the first node in the network [32,35,36]. The main energy consumption of the sensing nodes is while receiving and sending data packets, so reducing the number of aggregations can reduce energy consumption. The optimization performance in terms of lifetime is studied in this sub-section. The expectation of the maximum number of aggregations in the networks that have optimization effect are compared in Table 3. The AAR scheme decreases



the maximum number of aggregations in the network, thus postponing the death of the first node in the network. Thus, the lifetime of the network increases. There are optimization effects in different scenarios, and the effect increases with the increase of  $\lambda$  and  $P_a$ .

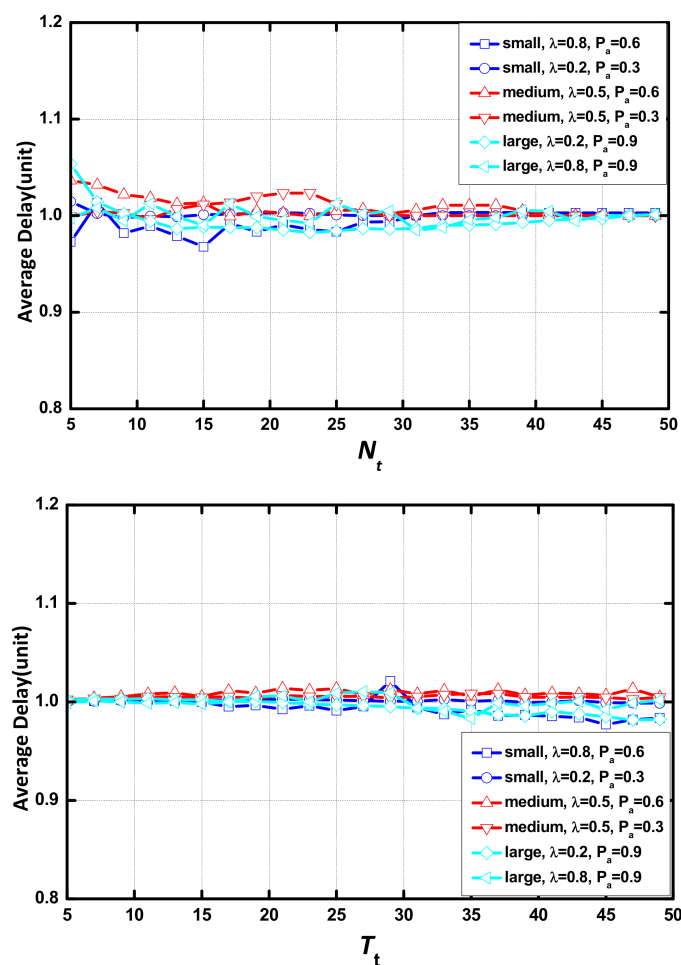
**Table 3.** Maximum of the aggregation in two schemes.

Scenario	Maximum of the Aggregation in CS	Maximum of the Aggregation in AAR	Ratio (%)
Small, (0.2,0.3)	590.58	479.18	81.13
Small, (0.8,0.6)	891.96	653.41	73.26
Medium, (0.5,0.3)	410.97	332.92	81.01
Medium, (0.5,0.6)	390.90	302.95	77.50
Large, (0.2,0.9)	94.24	76.84	81.54
Large, (0.8,0.9)	449.90	289.80	66.41

The average optimization performance of the AAR scheme reduces the average delay by 14.91% and the lifetime increases by 30.91% when  $N_t$  and  $T_t$  are the dominant parameters that determine the performance of the packet aggregation process.

#### 5.2.4. The Number of Aggregation of AAR Scheme

The optimization methods should improve the QoS by consuming limited extra energy. As revealed in Figure 15, the AAR scheme has a very limited impact on the number of aggregations. The ratio of the number of aggregations is approximately 1. This means that there is no extra energy consumption regarding aggregation in the AAR scheme.

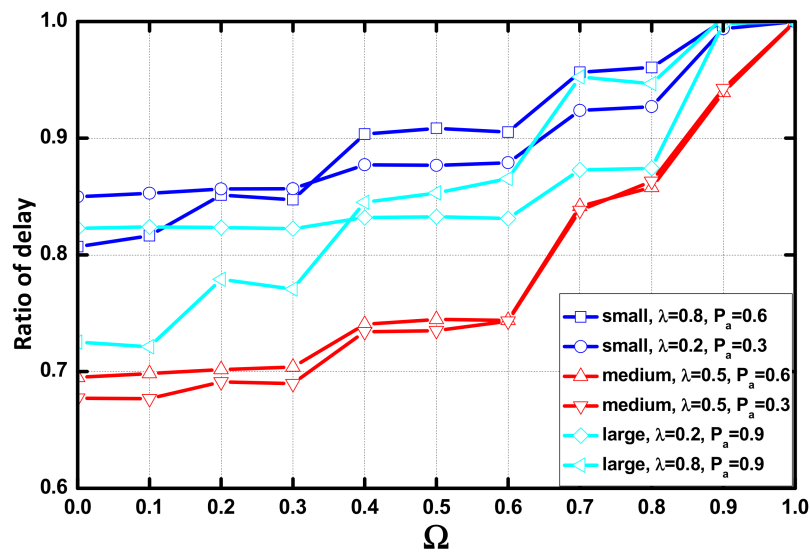


**Figure 15.** Ratio of the number of aggregations vs.  $N_t$  or  $T_t$ .

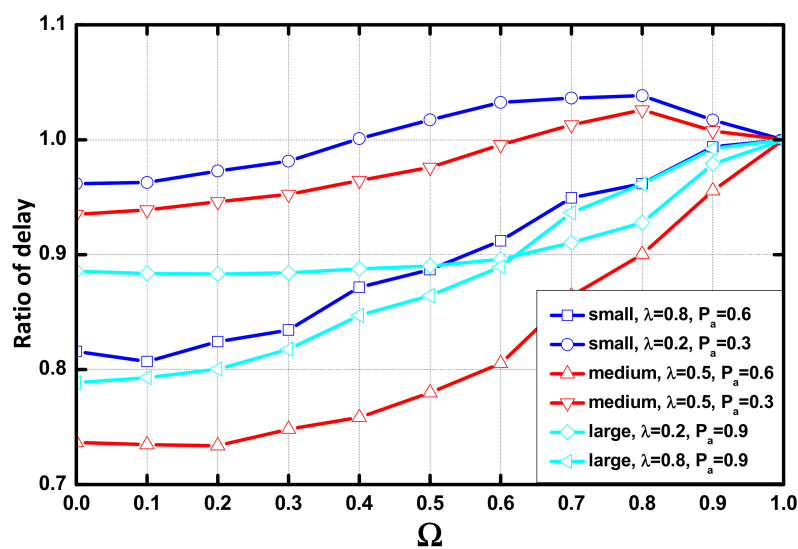
### 5.3. Performance of Optimizing at Different Node Selection Parameter

The node selection parameter  $\Omega$  determines the nodes in each layer that can receive the aggregation queues. In this sub-section, the parameter is set to 0, 0.1, 0.2,  $\dots$ , 1.0 (in the case of  $\Omega = 1.0$ , all of the aggregation queues are transmitted to the default parent node, that is, there is no optimization), we study the influence of different node selection parameters on the delay optimization.

In Figure 16a we find that the optimization performance decreases as the value of  $\Omega$  increases. The performance disappears when  $\Omega$  equals 1. When the value of  $N_t$  is 15, the optimization effect on the delay also decreases when  $\Omega$  increases. We notice that there is negative optimization when  $P_\alpha = 0.3$ . In the study in Section 5.2, we determine that the large  $N_t$  has no constraint on aggregation.  $T_t$  is the only parameter that controls the occurrence of aggregation. Therefore, there is no optimization in these environments.



(a)



(b)

Figure 16. Cont.

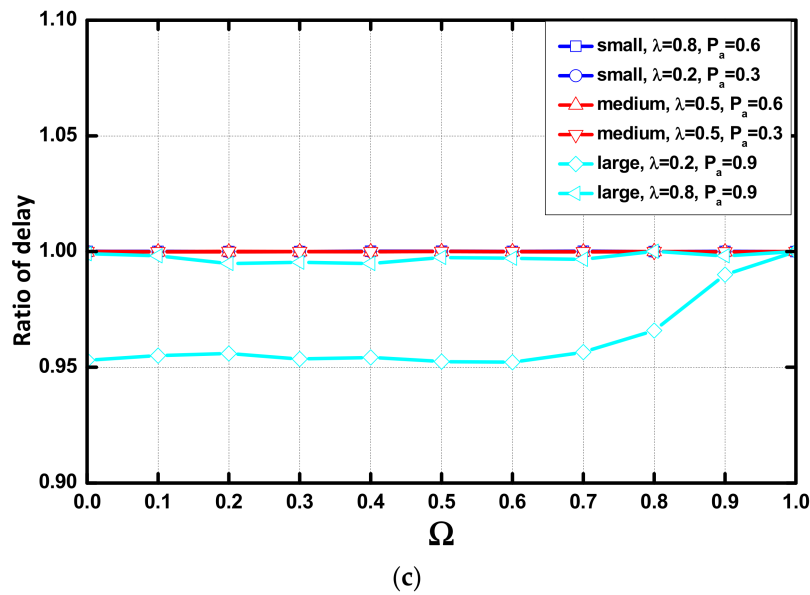


Figure 16. Ratio of delay vs.  $\Omega$  (a)  $N_t = 5$ ; (b)  $N_t = 15$ ; (c)  $N_t = 25$ .

Figure 16c shows that when  $N_t = 25$  and  $T_t = 25$ , there was no optimization except for the network whose  $\lambda = 0.8$  and  $P_\alpha = 0.9$ , as the area where the AAR works is large when the network has a large value of  $\lambda$  and  $P_\alpha$ . Therefore, performance are not affected by the value of  $\Omega$  except for the network with a large value of  $\lambda$  and  $P_\alpha$ .

#### 5.4. Performance of the AAR Scheme VS. the Improved AAR Scheme

Based on the AAR scheme, an improved AAR scheme (IMAAR) is proposed in this paper. It improves the QoS. The innovation of the IMAAR scheme is that because the data on the nodes far away from the sink use more hops to be transmitted to the sink and delay exists in every hop, delay is generated during the data collection process. Therefore, to reduce the delay, we need to reduce the delay in the process of data transmission from the nodes far from sink to the sink.

In this subsection, we propose a strategy to adjust the value of  $N_t$  and  $T_t$ . The expectation of the maximum number of aggregations in each layer of the network using the AAR scheme are calculated. In Figure 17, we see that the number of aggregations decreases with the increase of hops to the sink. Therefore, the nodes far from the sink have remaining energy when the first node dies. In the improved AAR scheme, the nodes far from the sink are set to a smaller value of aggregation timer ( $T_t$ ) and queue length threshold ( $N_t$ ) to reduce delay, so the residual energy can be fully used. The decrease in the number of polymerizations is influenced by  $\lambda$ .

Due to the homogeneity of the nodes, the upper limit of the number of aggregations  $\wp$  of all nodes in the network is equal. The number of aggregations performed in a node is  $\gamma$ . The energy efficiency  $\Xi$  can be expressed as Equation (11):

$$\Xi = \frac{\sum_{i=1}^m \sum_{j=1}^{n_i} \gamma_i^j}{\wp \mathcal{N}} \quad (11)$$

In the AAR scheme, the value of  $N_t$  and  $T_t$  are set by predetermined  $N_t(\overline{N_t})$  and predetermined  $T_t(\overline{T_t})$ . We design a formula to achieve the configuration of  $N_t$  and  $T_t$  for the node in layer  $i$ , which has shown below (the formula of  $T_t$  is same that of  $N_t$ ):

$$N_t = \begin{cases} \overline{N_t} \frac{L - i(\log_2 \frac{1}{\lambda})^3}{L}, & \lambda > 0.5 \\ \overline{N_t} \frac{L - i}{L} \log_2 \frac{1}{\lambda}, & \lambda \leq 0.5 \end{cases} \quad (12)$$

We ran the simulation to evaluate the energy efficiency. The upper bound of aggregation is 500. As long as the number of aggregations of one node reaches the upper limit, the network is considered dead. The number of aggregations of each node is counted to calculate the energy efficiency. The energy efficiency of the two schemes in different network scenarios and environments is seen in Table 4. This finding shows IMAAR significantly improves the energy efficiency of the network, which is improved by an average of 76.40%.

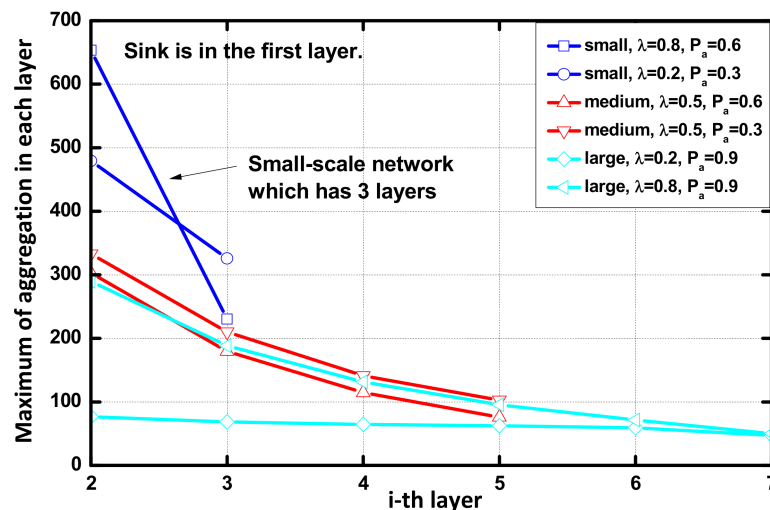


Figure 17. Maximum number of aggregations in each layer.

Table 4. Energy efficiency of two schemes.

Scenario	Energy Efficiency of AAR (%)	Energy Efficiency of IMAAR (%)
Small, (0.2,0.3)	76.06	90.80
Small, (0.8,0.6)	48.94	82.53
Medium, (0.5,0.3)	35.85	67.55
Medium, (0.5,0.6)	35.71	68.88
Large, (0.2,0.9)	73.88	79.14
Large, (0.8,0.9)	25.98	73.25

## 6. Conclusions

In this paper, we have presented an adaptive aggregation routing (ARR) scheme with the belief that our proposed scheme is an efficient means to reduce delay and improve the lifetime for WSNs. To address different applications, this scheme can be implemented in all networks with a tree topology. In this scheme, the node assignment algorithm (NAAL) is proposed for dynamically assigning the aggregation queues in the lower layer to the nodes in the upper layer. According to the state of the data queue in the nodes, this algorithm dynamically selects nodes that receive the aggregation queues in this cycle, while other nodes in the same layer can sleep to save energy. First, the nodes with a long data packet waiting queue are selected. Next, the priority is set according to the length of the queue in these nodes. Finally, the aggregation queues are assigned to these nodes by priority. In this process, the node is guaranteed to aggregate by receiving sufficient data. This increases the aggregation frequency and decreases the delay. Considering the aggregation threshold, the aggregation timer, the data aggregation ratio and probability that produce a data packet each cycle as the dominant parameters that determine the performance of the data aggregation and transmission process, we conducted simulation experiments in different cases. The performance of the proposed network scheme is compared with the common scheme that the aggregation queues only transmit to the parent node by default. The results are consistent with our expectations. The proposed network structure is shown to

be more efficient in terms of delay and lifetime during the data collection process, and this scheme does not cause extra energy consumption in aggregation. The AAR scheme works well when  $N_t$  has a strong constraint on delay, or the value of  $\lambda$  or  $P_\alpha$  is large, especially the value of  $P_\alpha$ . When  $T_t$  is the only parameter that operates the occurrence of aggregation, the AAR scheme does not work. In the network in which the AAR scheme has optimization effect, the delay is reduced by 14.91% and the lifetime is improved by 30.91%. The node selection parameter is designed to select the nodes with a long data queue. Through the simulations, we find that the small value of the parameter performs well regarding optimization, the optimum value of the parameter is 0. Based on the AAR scheme, an improved AAR scheme (IMAAR) is proposed in this paper. The IMAAR performs well regarding energy efficiency. Thus, the residual energy is fully used. In the experiments, IMAAR improves the energy efficiency by 76.40%. To the best of our knowledge, there is no optimization method based on the state of the data queue in the node. There is room for development for optimization based on a given state of the node. We expect that this study will contribute to the development of new optimization methods.

**Acknowledgments:** This work was supported in part by the National Natural Science Foundation of China (61772554, 61572435, 61572528), the Open Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (No. ICT1800391), the National Basic Research Program of China (973 Program) (2014CB046305), and the Natural Science Foundation of Zhejiang Province (LZ16F020001, LY17F020032).

**Author Contributions:** X.L. designed the algorithms and wrote part of the manuscript. A.L. conceived of the work, designed the algorithms, and wrote part of the manuscript. M.X., N.N.X., Z.Z. and Z.C. comment the paper.

**Conflicts of Interest:** The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

## References

1. Bhuiyan, M.Z.A.; Wang, G.; Wu, J.; Cao, J.; Liu, X.; Wang, T. Dependable structural health monitoring using wireless sensor networks. *IEEE Trans. Dependable Secur. Comput.* **2017**, *14*, 363–376. [CrossRef]
2. Wang, J.; Liu, A.; Yan, T.; Zeng, Z. A Resource Allocation Model Based on Double-sided Combinational Auctions for Transparent Computing. *Peer-to-Peer Netw. Appl.* **2017**. [CrossRef]
3. Xiao, F.; Liu, W.; Li, Z.; Chen, L.; Wang, R. Noise-tolerant Wireless Sensor Networks Localization via Multi-norms Regularized Matrix Completion. *IEEE Trans. Veh. Technol.* **2017**, 1–11. [CrossRef]
4. Bhuiyan, M.Z.A.; Wu, J.; Wang, G.; Wang, T.; Hassan, M.M. e-Sampling: Event-Sensitive Autonomous Adaptive Sensing and Low-Cost Monitoring in Networked Sensing Systems. *ACM Trans. Auton. Adapt. Syst.* **2017**, *12*, 1. [CrossRef]
5. Dai, H.; Chen, G.; Wang, C.; Wang, S.; Wu, X.; Wu, F. Quality of energy provisioning for wireless power transfer. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 527–537. [CrossRef]
6. Jiang, W.; Wang, G.; Bhuiyan, M.Z.A.; Wu, J. Understanding graph-based trust evaluation in online social networks: Methodologies and challenges. *ACM Comput. Surv. (CSUR)* **2016**, *49*, 10. [CrossRef]
7. Tang, J.; Liu, A.; Zhao, M.; Wang, T. An Aggregate Signature based Trust Routing for Data Gathering in Sensor Networks. *Secur. Commun. Netw.* **2018**. [CrossRef]
8. Huang, M.; Liu, A.; Wang, T.; Huang, C. Green Data Gathering under Delay Differentiated Services Constraint for Internet of Things. *Wirel. Commun. Mob. Comput.* **2018**. [CrossRef]
9. Wu, M.; Wu, Y.; Liu, X.; Ma, M.; Liu, A.; Zhao, M. Learning Based Synchronous Approach from Forwarding Nodes to Reduce the Delay for Industrial Internet of Things. *EURASIP J. Wirel. Commun. Netw.* **2018**, 10. [CrossRef]
10. Gui, J.S.; Hui, L.H.; Xiong, N.X. Enhancing Cellular Coverage Quality by Virtual Access Point and Wireless Power Transfer. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 9218239. [CrossRef]
11. Xin, H.; Liu, X. Energy-balanced transmission with accurate distances for strip-based wireless sensor networks. *IEEE Access* **2017**, *5*, 16193–16204. [CrossRef]
12. Zhao, S.; Liu, A. High Performance Target Tracking Scheme with Low Prediction Precision Requirement in WSNs. *Int. J. Ad. Hoc. Ubiquitous Comput.* **2017**. Available online: <http://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijahuc> (accessed on 30 March 2018).

13. Xie, K.; Wang, X.; Wen, J.; Cao, J. Cooperative routing with relay assignment in multiradio multihop wireless networks. *IEEE/ACM Trans. Netw. (TON)* **2016**, *24*, 859–872. [[CrossRef](#)]
14. Liu, A.; Huang, M.; Zhao, M.; Wang, T. A Smart High-Speed Backbone Path Construction Approach for Energy and Delay Optimization in WSNs. *IEEE Access* **2018**. [[CrossRef](#)]
15. Liu, X.; Zhao, S.; Liu, A.; Xiong, N.; Vasilakos, A.V. Knowledge-aware Proactive Nodes Selection Approach for Energy management in Internet of Things. *Future Gener. Comput. Syst.* **2017**. [[CrossRef](#)]
16. Ren, Y.; Liu, A.; Zhao, M.; Huang, C.; Wang, T. A Quality Utilization Aware based Data Gathering for Vehicular Communication Networks. *Wirel. Commun. Mob. Comput.* **2018**, 6353714. [[CrossRef](#)]
17. Liu, Y.; Liu, A.; Guo, S.; Li, Z.; Choi, Y.J. Context-aware collect data with energy efficient in Cyber-physical cloud systems. *Future Gener. Comput. Syst.* **2017**. [[CrossRef](#)]
18. Liu, Y.; Ota, K.; Zhang, K.; Ma, M.; Xiong, N.; Liu, A.; Long, J. QTSAC: A Energy efficient MAC Protocol for Delay Minimized in Wireless Sensor networks. *IEEE Access* **2018**. [[CrossRef](#)]
19. Tang, J.; Liu, A.; Zhang, J.; Zeng, Z.; Xiong, N.; Wang, T. A Security Routing Scheme Using Traceback Approach for Energy Harvesting Sensor Networks. *Sensors* **2018**, *18*, 751. [[CrossRef](#)] [[PubMed](#)]
20. Yang, G.; He, S.; Shi, Z. Leveraging crowdsourcing for efficient malicious users detection in large-scale social networks. *IEEE Internet Things J.* **2017**, *4*, 330–339. [[CrossRef](#)]
21. Ota, K.; Dong, M.; Gui, J.; Liu, A. QUOIN: Incentive Mechanisms for Crowd Sensing Networks. *IEEE Netw. Mag.* **2018**. [[CrossRef](#)]
22. Ma, F.; Liu, X.; Liu, A.; Zhao, M.; Huang, C.; Wang, T. A Time and Location Correlation Incentive Scheme for Deeply Data Gathering in Crowdsourcing Networks. *Wirel. Commun. Mob. Comput.* **2018**, 8052620. [[CrossRef](#)]
23. Sun, W.; Cai, Z.; Liu, F.; Fang, S.; Wang, G.; Li, Y. Data Processing and Text Mining Technologies on Electronic Medical Records: A Review. *J. Healthc. Eng.* **2018**, *2018*, 4302425. [[CrossRef](#)]
24. Huang, M.; Liu, Y.; Zhang, N.; Xiong, N.; Liu, A.; Zeng, Z.; Song, H. A Services Routing based Caching Scheme for Cloud Assisted CRNs. *IEEE Access* **2018**. [[CrossRef](#)]
25. Liu, Q.; Liu, A. On the hybrid using of unicast-broadcast in wireless sensor networks. *Comput. Electr. Eng.* **2017**. [[CrossRef](#)]
26. Teng, H.; Liu, X.; Liu, A.; Shen, H.; Huang, C.; Wang, T. Adaptive Transmission Power Control for Reliable Data Forwarding in Sensor based Networks. *Wirel. Commun. Mob. Comput.* **2018**, 2068375. [[CrossRef](#)]
27. Tan, J.; Liu, A.; Zhao, M.; Shen, H.; Ma, M. Cross Layer Design for Reducing Delay and Maximizing Lifetime in Industrial Wireless Sensor Networks. *EURASIP J. Wirel. Commun. Netw.* **2018**. [[CrossRef](#)]
28. Xie, K.; Cao, J.; Wang, X.; Wen, J. Optimal resource allocation for reliable and energy efficient cooperative communications. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 4994–5007. [[CrossRef](#)]
29. Kim, U.H.; Kong, E.; Choi, H.H.; Lee, J.R. Analysis of Aggregation Delay for Multisource Sensor Data with On-Off Traffic Pattern in Wireless Body Area Networks. *Sensors* **2016**, *16*, 1622. [[CrossRef](#)] [[PubMed](#)]
30. Pu, L.; Chen, X.; Xu, J.; Fu, X. D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3887–3901. [[CrossRef](#)]
31. Xu, J.; Liu, A.; Xiong, N.; Wang, T.; Zuo, Z. Integrated Collaborative Filtering Recommendation in Social Cyber-Physical Systems. *Int. J. Distrib. Sens. Netw.* **2017**, *13*, 1550147717749745. [[CrossRef](#)]
32. Liu, X.; Xiong, N.; Zhang, N.; Liu, A.; Shen, H.; Huang, C. A Trust with Abstract Information Verified Routing Scheme for Cyber-physical Network. *IEEE Access* **2018**. [[CrossRef](#)]
33. Le Nguyen, P.; Ji, Y.; Liu, Z.; Vu, H.; Nguyen, K.V. Distributed hole-bypassing protocol in WSNs with constant stretch and load balancing. *Comput. Netw.* **2017**, *129*, 232–250. [[CrossRef](#)]
34. Dai, H.; Liu, Y.; Chen, G.; Wu, X.; He, T.; Liu, A.X.; Ma, H. Safe charging for wireless power transfer. *IEEE/ACM Trans. Netw.* **2017**, *25*, 3531–3544. [[CrossRef](#)]
35. Yadav, P.; McCann, J.A.; Pereira, T. Self-Synchronization in Duty-cycled Internet of Things (IoT) Applications. *IEEE Internet Things J.* **2017**, *4*, 2058–2069. [[CrossRef](#)]
36. Harb, H.; Makhoul, A.; Laiymani, D.; Jaber, A. A Distance-Based Data Aggregation Technique for Periodic Sensor Networks. *ACM Trans. Sens. Netw. (TOSN)* **2017**, *13*, 32. [[CrossRef](#)]
37. Randhawa, S.; Jain, S. Data Aggregation in Wireless Sensor Networks: Previous Research, Current Status and Future Directions. *Wirel. Pers. Commun.* **2017**, *97*, 3355–3425. [[CrossRef](#)]



38. Villas, L.A.; Boukerche, A.; Ramos, H.S.; de Oliveira, H.A.F.; de Araujo, R.B.; Loureiro, A.A.F. DRINA: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks. *IEEE Trans. Comput.* **2013**, *62*, 676–689. [\[CrossRef\]](#)
39. Liu, X.; Li, G.; Zhang, S.; Liu, A. Big Program Code Dissemination Scheme for Emergency Software-define Wireless Sensor Networks. *Peer-to-Peer Netw. Appl.* **2017**. [\[CrossRef\]](#)
40. Sun, W.; Cai, Z.; Li, Y.; Liu, F.; Fang, S.; Wang, G. Security and Privacy in the Medical Internet of Things. *Secur. Commun. Netw.* **2018**, *2018*, 5978636. [\[CrossRef\]](#)
41. Dong, M.; Ota, K.; Liu, A.; Guo, M. Joint optimization of lifetime and transport delay under reliability constraint wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 225–236. [\[CrossRef\]](#)
42. Park, J.; Lee, S.; Yoo, S. Time slot assignment for convergecast in wireless sensor networks. *J. Parallel Distrib. Comput.* **2015**, *83*, 70–82. [\[CrossRef\]](#)
43. Badreddine, W.; Khernane, N.; Potop-Butucaru, M.; Chaudet, C. Convergecast in wireless body area networks. *Ad Hoc Netw.* **2017**, *66*, 40–51. [\[CrossRef\]](#)
44. Liu, Y.; Liu, A.; Chen, Z. Analysis and improvement of send-and-wait automatic repeat-request protocols for wireless sensor networks. *Wirel. Pers. Commun.* **2015**, *81*, 923–959. [\[CrossRef\]](#)
45. Huang, S.; Wan, P.; Vu, C.T.; Li, Y.; Yao, F. Nearly constant approximation for data aggregation scheduling in wireless sensor networks. In Proceedings of the 26th IEEE international conference on computer communications (INFOCOM 2007), Vancouver, BC, Canada, 31 July–3 August 2007; pp. 366–372.
46. Xu, X.H.; Li, X.Y.; Mao, X.F.; Tang, S.; Wang, S. A delay-efficient algorithm for data aggregation in multihop wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2011**, *22*, 163–175.
47. Ez-zazi, I.; Arioua, M.; El Oualkadi, A.; Lorenz, P. On the performance of adaptive coding schemes for energy efficient and reliable clustered wireless sensor networks. *Ad Hoc Netw.* **2017**, *64*, 99–111. [\[CrossRef\]](#)
48. Liu, X. Node Deployment Based on Extra Path Creation for Wireless Sensor Networks on Mountain Roads. *IEEE Commun. Lett.* **2017**, *21*, 2376–2379. [\[CrossRef\]](#)
49. Gui, J.; Deng, J. Multi-hop Relay-Aided Underlay D2D Communications for Improving Cellular Coverage Quality. *IEEE Access* **2018**, *6*, 14318–14338. [\[CrossRef\]](#)
50. Chen, Z.; Liu, A.; Li, Z.; Choi, Y.J.; Li, J. Distributed duty cycle control for delay improvement in wireless sensor networks. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 559–578. [\[CrossRef\]](#)
51. Chen, X.; Pu, L.; Gao, L.; Wu, W.; Wu, D. Exploiting massive D2D collaboration for energy-efficient mobile edge computing. *IEEE Wirel. Commun.* **2017**, *24*, 64–71. [\[CrossRef\]](#)
52. Liu, X.; Liu, Y.; Xiong, N.; Zhang, N.; Liu, A.; Shen, H.; Huang, C. Construction of Large-scale Low Cost Deliver Infrastructure using Vehicular Networks. *IEEE Access* **2018**. [\[CrossRef\]](#)
53. Naranjo, P.G.V.; Shojafar, M.; Mostafaei, H.; Pooranian, Z.; Baccarelli, E. P-SEP: A prolong stable election routing algorithm for energy-limited heterogeneous fog-supported wireless sensor networks. *J. Supercomput.* **2017**, *73*, 733–755. [\[CrossRef\]](#)
54. Huang, M.; Liu, A.; Zhao, M.; Wang, T. Multi Working Sets Alternate Covering Scheme for Continuous Partial Coverage in WSNs. *Peer-to-Peer Netw. Appl.* **2018**. [\[CrossRef\]](#)
55. Nazhad, S.H.H.; Shojafar, M.; Shamshirband, S.; Conti, M. An efficient routing protocol for the QoS support of large-scale MANETs. *Int. J. Commun. Syst.* **2018**, *31*, 1–18. [\[CrossRef\]](#)
56. Li, J.; Liu, Z.; Chen, X.; Xhafa, F.; Tan, X.; Wong, D.S. L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing. *Knowl. Based Syst.* **2015**, *79*, 18–26. [\[CrossRef\]](#)
57. Xu, Q.; Su, Z.; Zheng, Q.; Luo, M.; Dong, B. Secure Content Delivery with Edge Nodes to Save Caching Resources for Mobile Users in Green Cities. *IEEE Trans. Ind. Inf.* **2017**. [\[CrossRef\]](#)
58. Zhu, H.; Xiao, F.; Sun, L.; Wang, R.; Yang, P. R-TTWD: Robust device-free through-the-wall detection of moving human with WiFi. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1090–1103. [\[CrossRef\]](#)
59. Li, J.; Li, Y.K.; Chen, X.; Lee, P.P.C.; Lou, W. A hybrid cloud approach for secure authorized deduplication. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1206–1216. [\[CrossRef\]](#)
60. Chen, X.; Li, J.; Weng, J.; Ma, J.; Lou, W. Verifiable computation over large database with incremental updates. *IEEE Trans. Comput.* **2016**, *65*, 3184–3195. [\[CrossRef\]](#)
61. Li, J.; Li, J.; Chen, X.; Jia, C.; Lou, W. Identity-based encryption with outsourced revocation in cloud computing. *IEEE Trans. Comput.* **2015**, *64*, 425–437. [\[CrossRef\]](#)
62. Liu, Q.; Wang, G.; Liu, X.; Peng, T.; Wu, J. Achieving reliable and secure services in cloud computing environments. *Comput. Electr. Eng.* **2017**, *59*, 153–164. [\[CrossRef\]](#)

63. Liu, Q.; Guo, Y.; Wu, J.; Wang, G. Effective Query Grouping Strategy in Clouds. *J. Comput. Sci. Technol.* **2017**, *32*, 1231–1249. [[CrossRef](#)]
64. Xie, M.; Bhanja, U.; Shao, J.; Zhang, G.; Wei, G. LDSCD: A loss and DoS resistant secure code dissemination algorithm supporting multiple authorized tenants. *Inf. Sci.* **2017**, *420*, 37–48. [[CrossRef](#)]
65. Liu, A.; Chen, W.; Liu, X. Delay Optimal Opportunistic Pipeline Routing Scheme for Cognitive Radio Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2018**. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).