

Research Article

Delay-Aware Program Codes Dissemination Scheme in Internet of Everything

Yixuan Xu,¹ Anfeng Liu,¹ and Changqin Huang^{2,3}

¹*School of Information Science and Engineering, Central South University, Changsha 410083, China*

²*School of Information Technology in Education, South China Normal University, Guangzhou 510631, China*

³*Beijing Hetian Yuxiang Internet Technology Co., Ltd., Beijing 100036, China*

Correspondence should be addressed to Anfeng Liu; afengliu@mail.csu.edu.cn

Received 21 September 2016; Revised 5 November 2016; Accepted 16 November 2016

Academic Editor: Qingchen Zhang

Copyright © 2016 Yixuan Xu 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.

Due to recent advancements in big data, connection technologies, and smart devices, our environment is transforming into an “Internet of Everything” (IoE) environment. These smart devices can obtain new or special functions by reprogramming: upgrade their soft systems through receiving new version of program codes. However, bulk codes dissemination suffers from large delay, energy consumption, and number of retransmissions because of the unreliability of wireless links. In this paper, a delay-aware program dissemination (DAPD) scheme is proposed to disseminate program codes with fast, reliable, and energy-efficient style. We observe that although total energy is limited in wireless sensor network, there exists residual energy in nodes deployed far from the base station. Therefore, DAPD scheme improves the performance of bulk codes dissemination through the following two aspects. (1) Due to the fact that a high transmitting power can significantly improve the quality of wireless links, transmitting power of sensors with more residual energy is enhanced to improve link quality. (2) Due to the fact that performance of correlated dissemination tends to degrade in a highly dynamic environment, link correlation is autonomously updated in DAPD during codes dissemination to maintain improvements brought by correlated dissemination. Theoretical analysis and experimental results show that, compared with previous work, DAPD scheme improves the dissemination performance in terms of completion time, transmission cost, and the efficiency of energy utilization.

1. Introduction

Due to recent advancements in big data, connection technologies, and smart devices, the number of connected devices has already exceeded the number of people on Earth since 2011. Connected smart devices have reached 9 billion and are expected to grow more rapidly and reach 24 billion by 2020 [1]. Our environment is transforming into an “Internet of Everything” (IoE) environment. In 2012, global commercialization of IoT-based application systems generated a revenue of \$4.8 trillion [2]. Cisco estimates that, due to IoT, the global corporate profits will also increase approximately by 21% [3]. Due to extremely low costs of sensors and actuators, they can surely find their places in a wide range of applications in smart factory, smart city, and smart life, which lead to “Internet of Everything” (IoE) [4–8].

For example, many smart wireless sensors have been deployed in smart factory to monitor states of machines, sensing temperature, humidity, and sound [9, 10]. Wireless sensors are well suited for complicated industry environment because the deployment of them requires no wiring, so they have already been widely used in industrial production fields. Smart factory, which is composed of smart wireless sensors, can collect various kinds of data from machines and mine these collected data (i.e., industrial big data) to obtain valuable information for factory operation [11]. Machines are automatically controlled by obtained information to make an efficient production line (i.e., adequate production speed, low power consumption, and failure prediction). Therefore, smart wireless sensors make it possible to optimize the factory operation without human resources.

A sensor will work for several months or years once it is deployed [12]. However, in order to gain new functions, the upgrade of industrial production line requires sensors to upgrade simultaneously. One method called reprogramming is considered to be economic and convenient for such operations [13–16]. Besides, sometimes even without the upgrade of manufacturing facilities, these sensors will also need to upgrade to adapt to changes on production requirements. Therefore, in “Internet of Everything” (IoE) environment (e.g., smart factory, smart city, and smart life), it is common to disseminate new program codes to all wireless sensors through wireless communication. In this paper, such operation is called codes dissemination. Codes dissemination is a significant and crucial technique when sensors are deployed in environments where physically operating and reprogramming them are difficult or unfeasible. As a basic operation to enable wireless reprogramming, it attracts many research attentions in recent years. However, codes dissemination faces many challenges. First, the length of program codes is longer than the length of code packets and a network may include thousands of sensor nodes. Thus, disseminating large size codes correctly to a great amount of sensor nodes is one challenging issue. Another issue is dissemination delay (i.e., dissemination completion time, DCT), which refers to the required time for disseminating codes to all sensor nodes. It is better to obtain less dissemination completion time because large dissemination completion time may cause codes of different nodes to be inconsistent, resulting in loss of application due to the chaos about communication and signal transmission. Third, it is important to ensure complete reliability, which means that each active node in the network should receive program codes completely and correctly. Thus, large-scale programming codes dissemination for “Internet of Everything” (IoE) environment is a challenging task.

There are mainly three kinds of codes dissemination schemes. (1) The first one is a scheme called deluge [17]; this scheme uses negotiation to improve the performance of reliability. The method used in this scheme can be divided into three stages: broadcast, request, and send. Due to the fact that it needs three operations for each transmission, this scheme costs much to ensure the reliability and transmission delay. (2) The second scheme is flooding-based dissemination scheme [18]; this scheme removes request stage, which can make the speed for spreading program codes faster. However, the disadvantage is to cause broadcast storm problem. (3) The third scheme is link correlation-aware data dissemination scheme, which is proposed in [19]. The main idea of correlated dissemination (CD) is disseminating codes to the whole network by the broadcasting of sensor nodes; thus it is a one-to-many operation in unreliable wireless networks. Nodes which start broadcasting are called parent nodes and nodes which receive codes are called child nodes. In CD scheme, each node can only choose one node as its parent node and this parent node will take the responsibility of broadcasting codes to all its child nodes. Link correlation refers to the proportion of packets that are successfully (or unsuccessfully) received by all child nodes during broadcasting. Assigning sensor nodes with high link correlation to a same parent node can make retransmission packets more likely to be

needed by more than one child node and therefore number of retransmissions, dissemination completion time, and energy consumption are reduced. The main innovative idea of CD scheme is successfully building up a model to estimate link correlation and reducing number of retransmissions according to it.

Although many researches have already been done, some problems deserve further study [20]. (1) The first problem is the problem on improving the reliability of wireless link. Previous researches normally ensure the reliability of codes dissemination through multiple retransmissions in network layer. However, such solutions also bring problems on increasing the delay of codes dissemination and energy consumption, especially in a network with high packet loss ratio. Therefore, how to ensure link reliability and maintain low delay simultaneously is an important and challenging issue. (2) Although correlated dissemination (CD) is able to reduce retransmission packets, it faces the problem of sending extra packets to obtain link correlation, which consumes more energy of sensors and shortens the lifetime of network. Besides, link correlations tend to change dynamically during real-world codes dissemination. Therefore, how to overcome the extra energy consumption of sensors and obtain the latest link correlation simultaneously is another big concern.

Based on the analysis above, a delay-aware program dissemination (DAPD) scheme is proposed to disseminate program codes with fast, reliable, and energy-efficient style. The improvement of DAPD on the performance of codes dissemination is founded on two facts in wireless sensor network. (1) Link quality is related to transmitting power of sensor nodes directly. The former one will improve greatly when enhancing transmitting power, which leads to a decrease in the number of retransmissions and DCT. Besides, energy consumption on retransmitting codes also reduces. On the other hand, although total energy of sensors is limited, sensors are actually in a state of sensing machines during most of time, transmitting collected data to the base station by multihop. With this many-to-one operation, sensors around the base station not only need to transmit their own data but also take the responsibility of forwarding data originated from sensors far from the base station (far nodes). Therefore, energy consumption of these sensors is much larger than far nodes and residual energy will accumulate in far nodes during this period. If such residual energy can be exploited when disseminating codes, the performance will improve greatly. (2) The premise for correlated dissemination is using extra packets to obtain link correlation before disseminating codes. If the number of these packets is too small, link correlation cannot be obtained correctly and adequately. On the other hand, if it is too large, the lifetime of network will be shortened due to much extra energy consumption. So sensor nodes far from the base station (also with excess energy) are able to obtain link correlations adequately with our scheme to achieve the goal that the lifetime of network will not be influenced while dissemination performance is improved. The main contributions of the DAPD scheme are listed as follows.

(1) DAPD scheme enhances transmitting power of sensors with excess energy to improve link quality and reduce

dissemination completion time under the premise of not shortening the lifetime of network.

(2) DAPD scheme takes full advantage of residual energy to obtain link correlations correctly and adequately before codes dissemination. Besides, it also proposes a dynamic parent node selection algorithm during codes dissemination to quickly adapt to the latest environment and make use of link correlation more effectively.

(3) Through our theoretical analysis and simulation study, we demonstrate that, for DAPD scheme, codes dissemination completion time can be reduced and energy utilization efficiency can be enhanced simultaneously. Compared with former schemes, codes dissemination completion time can be reduced by as much as 19.05% (larger when the environment is highly dynamic). More importantly, the proposed scheme improves the performances without harming network lifetime, which is difficult to achieve in previous schemes.

The rest of this paper is organized as follows: Section 2 reviews related work. System models and problem statements are introduced in Section 3. In Section 4, a novel DAPD scheme is presented to disseminate program codes with fast, reliable and energy-efficient style. Performance analysis for DAPD scheme is provided in Section 5. Experimental results and comparisons are given in Section 6. Section 7 concludes the paper.

2. Related Work

Many program codes dissemination schemes have been proposed [21–28], with each of them focusing on one or two specific challenges during codes dissemination phase (e.g., latency, energy consumption, and reliability). Generally, these schemes can be divided into the following types based on their design purposes and requirements.

(1) The first type is schemes focusing on reducing latency. The objective of such schemes is to ensure the reliability of codes dissemination and reduce dissemination completion time simultaneously. Deluge can be considered as an example of such schemes [17]. Three-way handshake and ACK-based protocol are adopted in deluge for reliability. Besides, transmission delay can be improved through dividing codes into fixed size pages. In deluge, each node will advertise about local pages. When one node (receiver) learns that another node (sender) has pages not successfully received by itself, it will send a request to the sender and prepare to receive pages. Many other schemes are based on deluge. For example, rateless deluge reduces latency further through using random linear codes to encode packets [21].

Zheng et al. proposed Survival of the Fittest (SurF) to solve the problem that negotiations between sensor nodes tend to incur long dissemination completion time [23]. This scheme achieves a tradeoff between negotiation and flooding, which is another dissemination scheme but is considered to be energy-consuming. SurF selectively adopts two schemes (negotiation and flooding) to reduce dissemination completion time.

(2) The second type is schemes focusing on reducing energy consumption. The objective of such schemes is to

prolong the lifetime of wireless sensor network. In many energy-efficient schemes, sensor nodes alternate between active state and dormant state to reduce energy consumption [24, 25, 28, 29]. Kulkarni and Wang proposed one scheme called MNP after finding that one main source of energy consumption in deluge results from high degree of message collision [26]. In MNP, a sender selection algorithm is used to solve message collision problem. Besides, one node will go into dormant state if its neighbor nodes are transmitting packets already owned by itself. Due to a decrease on active radio time, energy consumption is significantly reduced. In addition, experiences show that single-hop reprogramming may achieve a better performance on dissemination completion time and energy consumption than multihop reprogramming under certain conditions. Therefore, one scheme called DStream is proposed [27], which has the abilities on both single-hop and multihop dissemination.

(3) Another kind of special dissemination scheme proposed recently is called link correlation-aware data dissemination scheme [19]. Due to the fact that codes dissemination is one kind of wireless broadcast in this scheme, disseminated codes will be received by more than one sensor in the broadcast domain. Basically, sensors with high link correlation are more likely to successfully (or unsuccessfully) receive packets with same packet ID. On the other hand, lost packets of sensors with low link correlation tend to be different from each other. Therefore, this scheme combines sensors with high correlation together, which makes retransmission packets have more possibilities to be needed by more than one sensor. Correlated dissemination performs well in terms of latency, energy consumption, and reliability when the environment of network is stable.

3. System Models and Problem Statements

3.1. Network Model. The network model that we adopt is shown in Figure 1, which can also be found in [9]. The whole network is composed of one base station and many sensor nodes evenly distributed in the network. The energy of base station is considered to be infinite. In contrast, sensor nodes are powered by batteries and total energy is limited.

Two main functions of network include data collection and codes dissemination. During the first operation, all sensor nodes need to transmit collected data back to base station by multihop with unicast style. Due to the unreliability of link, sensors may have to transmit data repeatedly. Packets transmitted during the first operation are called data packets in our scheme. The base station will transmit code packets to all sensor nodes in the second phase and sensor nodes will also participate in transmitting these code packets through broadcast style because of the limited transmission radius of base station.

3.2. Link Quality Model. In this section, the relation between link quality q and transmitting power P_t is introduced. Specifically, link quality is measured by packet reception rate (PRR).

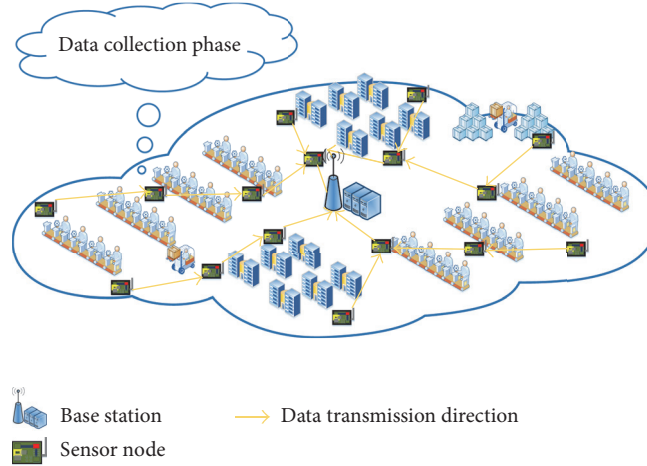


FIGURE 1: Industrial wireless sensor network.

In [30], Zuniga and Krishnamachari analyzed parameters on channel and proposed a mathematical formula to calculate packet reception rate (PRR):

$$q = \left(1 - \frac{1}{2} \exp^{-\gamma/2(B_N/R_D)}\right)^{8f}, \quad (1)$$

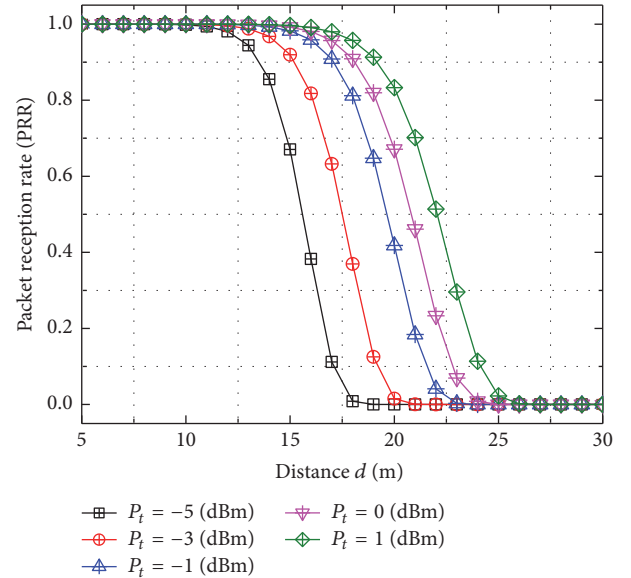
where R_D is data rate in bits, B_N is the noise bandwidth, and f is the frame size. SNR (signal-to-noise ratio) γ can be calculated through (2) if given specific transmitting power P_t and distance between transmitter and receiver d .

$$\gamma_{\text{dB}}(d) = P_t - \text{PL}(d_0) - 10n \log_{10}\left(\frac{d}{d_0}\right) - N(0, \sigma) - P_n, \quad (2)$$

where d_0 is a reference distance and n is the path loss exponent. Its value can be set to 2–4 if transmission approximately follows free space model. $\text{PL}(d_0)$ and $10n \log_{10}(d/d_0)$ indicate attenuation caused by the adopted log-normal shadowing path loss model [31, 32]. In detail, path loss is influenced by signal diffusion and characteristics of channel while shadowing effect is caused by obstacles between transmitter and receiver [33]. $N(0, \sigma)$ is a zero-mean Gaussian RV with standard derivation σ and P_n is noise floor. Their value can be obtained through empirical measurements. Curves in Figure 2 show relation between packet reception rate and d under different transmitting power P_t in a static environment ($\sigma = 0$).

3.3. Link Correlation Model. Link correlation model is used to obtain link correlation before codes dissemination phase. In this model, link correlation is obtained through broadcasting HELLO messages and reception vectors are used to keep information on receptions. One simple example shown in Figure 3 is used to demonstrate the construction of reception vectors and the calculation of link correlation.

First, nodes A and B broadcast 10 HELLO messages to one-hop downstream sensors in their broadcast domain

FIGURE 2: Relation between PRR and distance d ($R_D = 19.2$ kbps, $B_N = 30$ kHz, and $n = 4$).

(C, D, and E) separately. After successfully receiving one of these HELLO messages, C, D, and E will reply with an ACK (ACKnowledgement sent by receivers to confirm that data has been received successfully). Second, A and B construct reception vectors for C, D, and E according to these ACKs: if A receives the ACK for the i th HELLO message from C, then the i th element in the reception vector that A constructs for C will be 1. Otherwise, the i th element will be 0 because of the loss of HELLO message or ACK during transmission. Link reliability is not guaranteed in these two steps in order to reflect link correlation correctly (link reliability indicates that all packets can be received successfully through schemes like retransmission). Third, A and B will broadcast packets that contain information on these reception vectors to C,

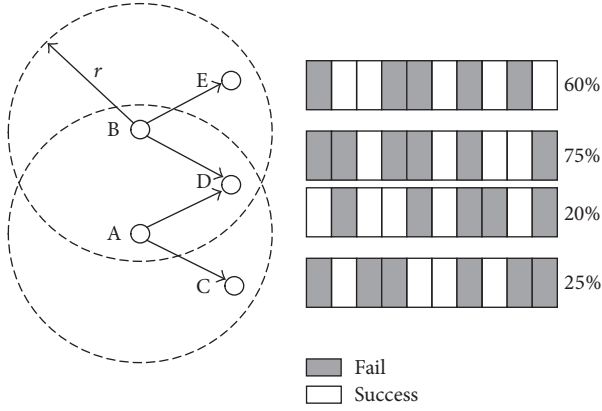


FIGURE 3: The illustration of link correlation.

D, and E. Upon receiving these packets, C, D, and E will use (3) to calculate link correlation C between themselves and A and B.

$$C(i, k) = \frac{\sum_{m=1}^N V_1(m) \times V_2(m) \times \cdots \times V_X(m)}{\sum_{m=1}^N V_k(m)}, \quad (3)$$

where N is the length of reception vector (also the number of HELLO messages), $V_k(m)$ is the m th element in the reception vector of node k , and i is the ID of node that broadcasts HELLO messages. In particular, $V_1(m) \times V_2(m) \times \cdots \times V_X(m)$ denotes the AND result of m th elements in reception vectors of all one-hop downstream nodes in the broadcast domain, which indicates that only when a HELLO message is received by all one-hop downstream nodes will the link correlation increase. For example, after A and B construct reception vectors for C and D and D and E separately and broadcast them, D will receive the reception vectors of C, D, and E as its location is covered by broadcast domains of A and B simultaneously. Then D uses (3) to calculate link correlations $C(A, D)$ and $C(B, D)$. Specifically, $C(A, D)$ is $1/5 = 20\%$, while $C(B, D)$ is $3/4 = 75\%$.

3.4. Energy Consumption Model. Two important sources of energy consumption are transmitting and receiving data. Their energy cost E_t and E_r can be estimated as follows:

$$\begin{aligned} E_t &= P_t \times \frac{D^t}{R_D}, \\ E_r &= P_r \times \frac{D^r}{R_D}, \end{aligned} \quad (4)$$

where P_t and P_r are transmitting power and receiving power separately. D^t and D^r denote data size that needs to be transmitted and received, and R_D is data rate.

3.5. Problem Statements. Delay-aware program dissemination (DAPD) focuses on reducing transmission delay during disseminating codes under the premise that the lifetime of network will not be influenced. Therefore, dissemination completion time (DCT) and residual energy in sensors are two main concerns. Problem statements are as follows.

(1) *To Minimize the Dissemination Completion Time.* Codes dissemination phase starts from the base station broadcasting code packets and ends until all active sensors in the network receive codes correctly. One aim of DAPD is to minimize time spent between these two time points.

(2) *To Avoid Influencing the Lifetime of Network.* Since one important operation in DAPD is to enhance transmitting power by utilizing excess energy, an overuse will definitely reduce the lifetime of network. Therefore, DAPD is designed based on two principles. (1) Available residual energy during codes dissemination phase in sensors is the difference between residual energy of themselves E_i and minimum residual energy in the network E_{\min} before this phase starts. (2) Residual energy distribution should be as uniform as possible after codes dissemination phase. Two statements above can be expressed as follows:

$$\begin{aligned} \max(E_i - E_{\min}), \quad i = 1, 2, \dots, M, \\ \min\left(\sum_{i=2}^M (E_i - E_1)^2\right), \end{aligned} \quad (5)$$

where E_i is the residual energy of sensor node i , E_{\min} is minimum residual energy in the network before disseminating codes, M is the total number of active sensors, and E_1 is the left energy of sensor node i after codes dissemination phase.

4. Scheme Design

4.1. Motivation

4.1.1. Unbalanced Energy Distribution. When the wireless sensor network is in the phase of data collection, neighbor sensors of the base station need to forward data originated in sensors far from the base station apart from transmitting back their own data, which leads to an unbalanced distribution of data load and energy consumption. Figure 4 shows the unbalanced distribution of residual energy in sensors with different distance from the base station after data collection phase. Furthermore, the lifetime of wireless sensor network can be defined as time that the network goes through until any sensor runs out of its energy (the first failure) [34]. Therefore, far nodes tend to have much underutilized energy during the lifetime of wireless sensor network if no additional scheme is taken.

Studies show that such residual energy can take up more than half of total energy in the network [34–36]. Hence, we consider taking advantage of residual energy in far nodes to enhance transmitting power during codes dissemination, and such enhancement will lead to a better link quality. Figure 5 presents the improvement on link quality after enhancing transmitting power, where link quality improves greatly when transmitting power is enhanced from -4 (dBm) to 4 (dBm); therefore the performance of codes dissemination can be improved.

With a more reliable link, number of retransmission and transmission delays can be reduced. Figure 6 is an illustration of expected dissemination completion time (DCT) for one

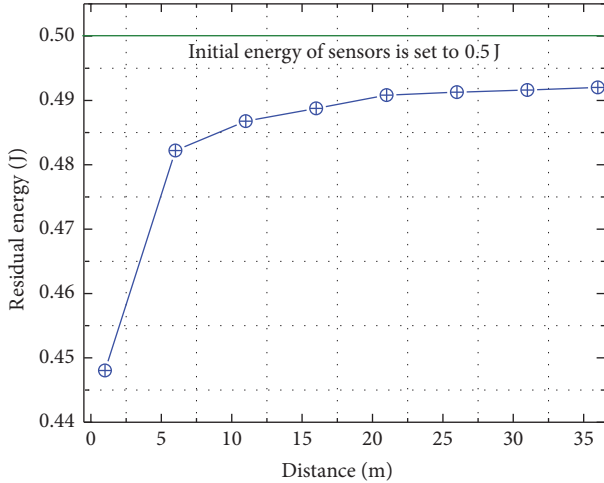


FIGURE 4: Distribution of residual energy on sensors.

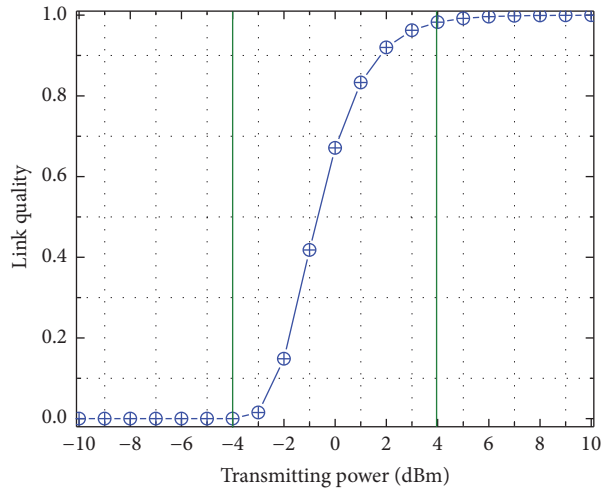


FIGURE 5: Relation between transmitting power and link quality ($R_D = 19.2$ kbps, $B_N = 30$ kHz, and $n = 4$).

sensor node to transmit code packets to its child nodes with different transmitting power.

Based on the analysis above, we can conclude that residual energy in sensors far from the base station can be used to enhance transmitting power during codes dissemination, and thus the performance of codes dissemination can be improved.

4.1.2. Change on Link Correlation. The utilization of link correlation during codes dissemination has already been proven to be fast and energy-efficient. However, experiments also show that the performance of correlated dissemination will degrade and be no better than deluge when in a highly dynamic environment. In detail, previous choices on parent nodes become outdated because link correlation will change over environment. The example in Figure 7 is used to illustrate the impact of changes on link correlation and the necessity on reselecting parent nodes.

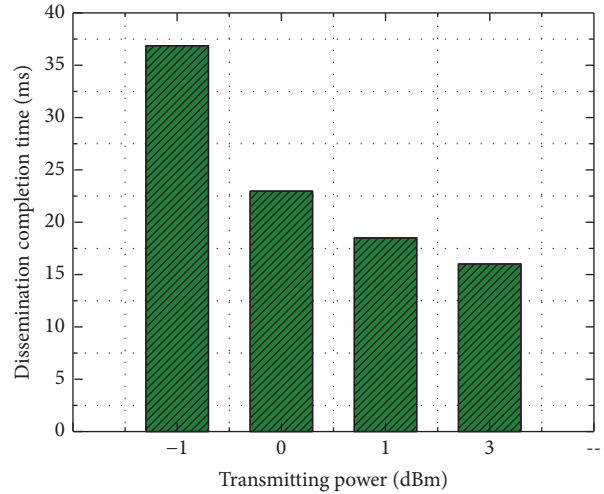


FIGURE 6: Expected dissemination completion time with different transmitting power.

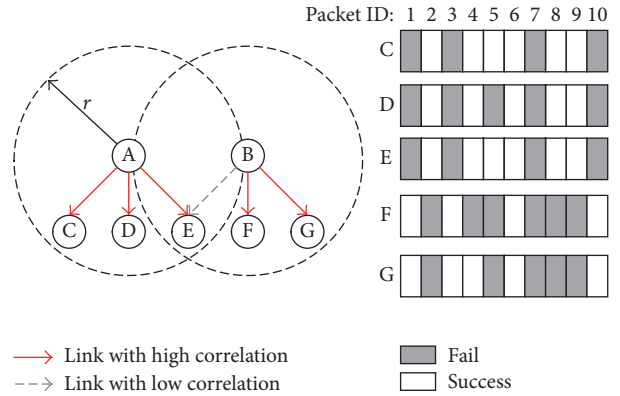


FIGURE 7: One example where A and B broadcast code packets to C, D, E, F, and G.

Initially, C, D, and E have already chosen A to be their parent node according to link correlation, while B has been chosen by F and G. After that, A and B start to broadcast code packets (10 packets one time) and will not broadcast following packets until these packets are correctly received by all their child nodes. Receptions on these packets are shown in the right part of Figure 7. The conclusion that link correlation between C, D, and E (or F and G) is high can be concluded from receptions. To simplify the illustration, we assume that all following packets retransmitted by A and B will be successfully received by C, D, E, F, and G. Therefore, A needs to retransmit 5 packets (ID: 1, 3, 5, 7, and 10) to C, D, and E, while B needs to retransmit 6 packets (ID: 2, 4, 5, 7, 8, and 9) to F and G.

Transmission on the next 10 packets will be similar to Figure 7 if the environment stands stable. However, this is not the case in the real world. One actual case after finishing transmitting the first 10 packets is shown in Figure 8.

The environment changes and link correlation $C(B, E)$ is now higher than $C(A, E)$, which can be seen from receptions on the following 10 code packets. If the parent node of C, D,

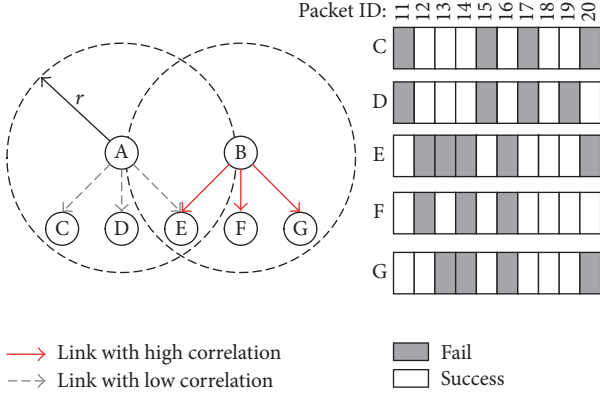


FIGURE 8: Link correlation changes after all child nodes correctly receive the first 10 code packets.

and E is still A, then A needs to retransmit 9 packets (ID: 11, 12, 13, 14, 15, 16, 17, 19, and 20). However, the performance will be improved if E reselects B as its parent node. Specifically, A will need to retransmit 5 packets (ID: 11, 15, 17, 19, and 20) to its child nodes (C and D), while B will need to retransmit 5 packets (ID: 12, 13, 14, 16, and 20) in this case. Hence, an operation that reselects parent nodes will lead to a better utilization of link correlation in the real world. Besides, compared with disseminating codes, cost on selecting parent node is relatively smaller. Therefore, it is possible to make better use of link correlation at the expense of going through another phase to reselect parent node when the environment is highly dynamic.

4.2. Design on Enhancing Transmitting Power. In order to enhance transmitting power without reducing the lifetime of wireless sensor network, we first need to correctly estimate residual energy in sensors before disseminating codes.

First, we analyze data load on each sensor during data collection phase. Timeout retransmission mechanism is adopted in our network model to ensure that all collected data can be sent back to base station correctly. Necessary notations are given as follows:

- i, j, k : sensor node ID
- h_i : hop count of node i
- P_i^t : transmitting power of i
- P_i^r : receiving power of i
- q_{ij} : link quality between i (transmitter) and j (receiver)
- N_i : number of packets that i needs to transmit to its one-hop upstream node
- S_{DATA} : size of data packets
- S_{ACK} : size of ACKs

Besides, as shown in Figure 9, i, j , and k are 3 neighbor sensors with different hop counts ($h_k = h_j + 1; h_j = h_i + 1$).

For example, j needs to receive N_k packets from k and transmit N_k ACKs back to k . Expected number of transmission is $1/q_{jk}q_{kj}$ because of the unreliability of link. On the

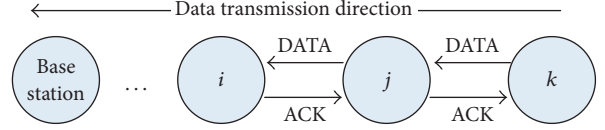


FIGURE 9: Three neighbor sensors, i, j , and k , with different hop counts during data collection phase.

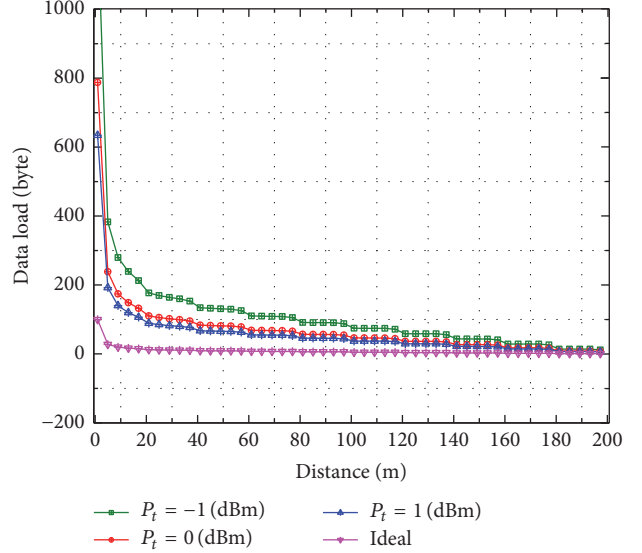


FIGURE 10: Data load on sensors with different distances from the base station.

other hand, j needs to transmit N_j packets to i and receive N_j ACKs from i and expected number of transmissions is $1/q_{ij}q_{ji}$. Therefore, we can obtain the amount of data that j needs to transmit D_j^t and receive D_j^r under the premise that N_k and N_j are known, which are shown in (6) and (7) separately. Besides, N_k and N_j can be calculated through (8) [37]. Equation (8) successfully estimates data load on sensor nodes with different distances from the base station in Send-Wait style with ACK protocol and no packet loss.

$$D_j^t = \frac{N_k}{q_{jk}q_{kj}} \times S_{\text{ACK}} + \frac{N_j}{q_{ij}q_{ji}} \times S_{\text{DATA}}, \quad (6)$$

$$D_j^r = N_k \times S_{\text{DATA}} + N_j \times S_{\text{ACK}}, \quad (7)$$

$$N_i = (z + 1) + \frac{z(z + 1)r}{2l}, \quad (8)$$

where l is the distance between i and the base station, r is the transmission radius, and z is the largest integer that satisfies $zr + l < R$ (radius of the network). Figure 10 shows data load on sensors with $r = 20$ m in a wireless sensor network with $R = 200$ m.

Second, residual energy in sensors E can be estimated according to the data load above, combined with energy consumption model in Section 3.4.

$$E = E_0 - n(E_t + E_r), \quad (9)$$

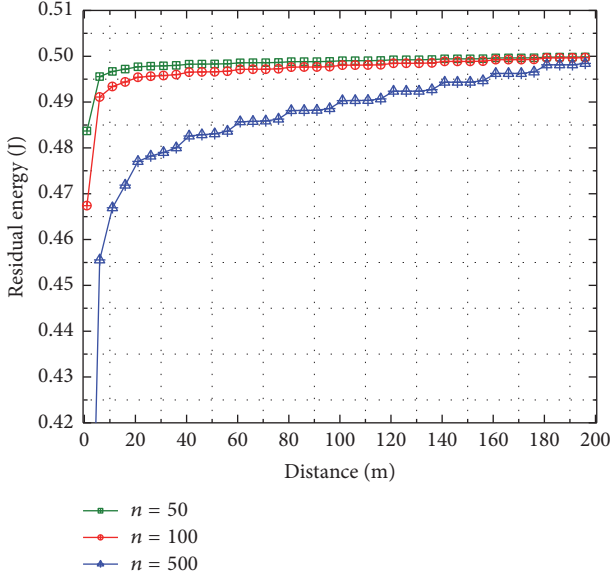


FIGURE 11: Residual energy in sensors after collecting data for different times.

where E_0 is initial energy in sensors and n is times of data collection. Figure 11 shows residual energy in sensors after collecting data for 50, 100, and 500 times.

To avoid the lifetime of network being influenced by the enhancement of transmitting power, an upper limit should be set according to sensors that have minimum energy left after data collection phase. From Figure 11, we can observe that such sensors tend to be deployed around the base station. Therefore, residual energy that can be used to enhance transmitting power E_a can be calculated through the following equation:

$$E_a = E_i - E_{\min}, \quad (10)$$

where E_i is the residual energy in sensor i and E_{\min} is minimum residual energy in the network. E_a will be used to enhance transmitting power during the phase that disseminates codes and the phase that reselects parent nodes. Therefore, an additional variable α is introduced in order to allocate E_a to these two phases properly. For example, enhanced transmitting power P'_t of sensor i during codes dissemination is calculated according to the following equation:

$$P'_t = P_t + \frac{\alpha(E_i - E_{\min})}{D} \times R_D \times \text{PRR}_0(P_t), \quad (11)$$

where D is total data size on code packets, P_t is the initial transmitting power, and $\text{PRR}_0(P_t)$ is the link quality under initial P_t . Due to the fact that the amount of data that needs to be transmitted in the phase that reselects parent node is smaller than that of codes dissemination phase, α should be larger than 0.5. Figure 12 shows the enhanced transmitting power of sensors in a case where D is 500 bytes and codes dissemination starts after collecting data for 50 times.

4.3. Our Methodology. In this section, we will show design details on delay-aware program dissemination (DAPD).

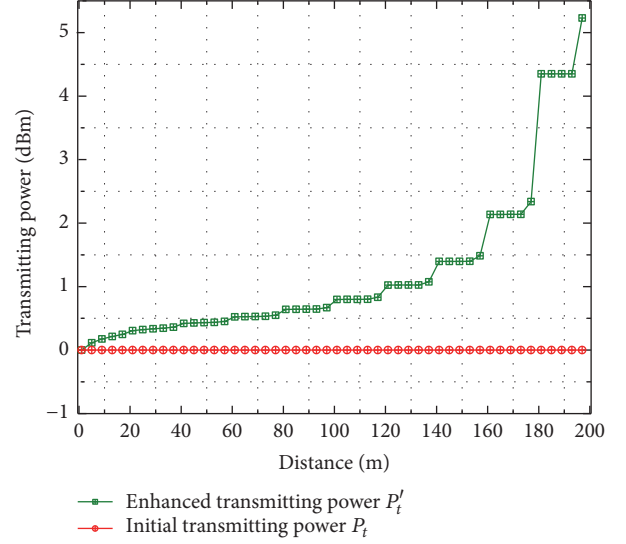


FIGURE 12: The enhancement of transmitting power.

4.3.1. Overview. Delay-aware program dissemination (DAPD) is one kind of bulk data dissemination and it has 3 salient characteristics. (1) It exploits excess energy of sensors to enhance transmitting power during codes dissemination. (2) It exploits link correlation to reduce dissemination completion time. (3) It selectively goes through fast parent node reselection phase to quickly adapt to changes on environment and recover the improvement brought by link correlation. DAPD is composed of three phases: initial parent node selection phase, codes dissemination phase, and fast parent node reselection phase. Following sections will show detailed information on these three phases.

4.3.2. Initial Parent Node Selection Phase. This phase is used to obtain link correlation and choose parent node according to link correlation before disseminating codes.

First, the base station will initiate a flooding which enables each sensor node to obtain its hop count. Second, each node broadcasts HELLO messages to all one-hop downstream nodes in its broadcast domain, which contain its own ID and hop count. These one-hop downstream nodes will reply with ACKs upon successfully receiving HELLO messages. Link reliabilities are not guaranteed here in order to reflect link correlation correctly. Third, nodes which broadcast HELLO messages construct reception vectors for one-hop downstream nodes. Detailed information on the construction of reception vector is shown in Section 3.3. At last, these reception vectors will be broadcast to one-hop downstream nodes and upon receiving packets that contain information on reception vectors, these one-hop downstream nodes will calculate link correlation between themselves and the transmitter according to (3).

Due to the fact that the location of these one-hop downstream nodes may be covered by more than one transmitters' broadcast domain, they may receive reception vectors from many transmitters. They will choose the transmitter with the highest link correlation to be their parent node and

send a CHOSEN message to inform this node. To ensure link reliability, timeout retransmission mechanism is adopted during transmission on reception vectors and CHOSEN message.

For example, C will receive reception vectors of itself and D from A in Figure 3. Therefore, C only needs to calculate link correlation for one time. ($C(A, C) = 1/4 = 25\%$) and it has to choose A as its parent node regardless of link correlation because C is only covered by the broadcast domain of A. However, D will receive reception vectors of all one-hop downstream nodes (C, D, and E). After using (3) to calculate link correlations between itself and A and B, it chooses B to be its parent node because $C(B, D)$ is much larger than $C(A, D)$.

4.3.3. Codes Dissemination Phase. After initial parent node selection phase, each sensor node in the network will obtain its parent node ID and all its child nodes ID. Next, codes dissemination phase is initiated by the base station broadcasting code packets. Sensor nodes also start broadcasting code packets to child nodes after successfully receiving all packets. During broadcasting, nodes will continuously broadcast N packets at a time and will not broadcast following packets until all child nodes' ACKs for these packets are received.

Besides, one node may receive packets from other one-hop upstream nodes apart from its parent node. In this case, it will compare the packet ID with N packet IDs that it can currently receive from its parent node. (1) The packet is not one of those N packets or has already been successfully received; then it will discard this packet. (2) The packet is one of those N packets and has not been received; then it will receive this packet and reply an ACK for this packet to its own parent node. Such mechanism will reduce dissemination completion time further. Code packets are transmitted according to operations above hop by hop until all sensor nodes receive codes successfully.

4.3.4. Fast Parent Node Reselection Phase. Section 4.1.2 shows that changes on environment during codes dissemination phase will lead to degradation on the performance of correlated dissemination. Current link correlation may be very different from the link correlation calculated before disseminating. Therefore, previous choices on parent node can be outdated. To reuse link correlation, another parent node selection phase is necessary. However, unlike Section 4.3.2 which initiates from the base station, a parent node selection phase that only happens between nodes and their one-hop downstream nodes is needed here. In detail, one sensor node will rebroadcast HELLO messages to all one-hop downstream nodes in its broadcast domain before transmitting code packets. Then, again, these one-hop downstream nodes will reply ACKs for each HELLO message. The following steps are same to initial parent node selection phase. Figure 13 shows the data transmission for one-hop downstream nodes ($i + 1$) to choose parent nodes (i), where V indicates packets that contain information on reception vectors; τ is extra time to make sure all potential ACKs can arrive.

Since the incentive for going through this reselection phase is to reuse link correlation, therefore, it is necessary for

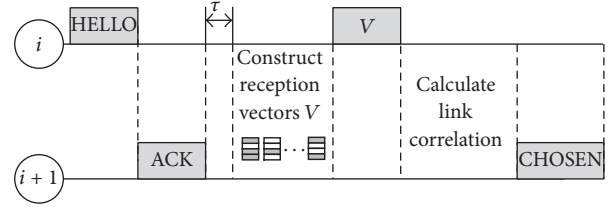


FIGURE 13: Data transmission during fast parent node reselection phase.

sensor nodes to keep monitoring on link correlation during codes dissemination phase. When link correlation drops below a predefined threshold, one sensor node can make the assumption that environment around itself has changed and a fast parent node reselection phase is needed before transmitting code packets to child nodes. However, it can be time-consuming and unreasonable to obtain link correlation through the same way described in Section 4.3.2 (needs parent node's participation). The method we adopted here to make quick estimations on link correlation is collecting statistics on the percent of unneeded retransmitted code packets. In detail, after finishing one round of transmission or retransmission, (12) is used to estimate link correlation:

$$s = 1 - \frac{k_u}{k_{\text{all}}}, \quad (12)$$

where k_u is the number of unneeded retransmitted code packets, k_{all} is the total number of received code packets, and s indicates the percent of useful retransmitted code packets for one sensor node. When s is lower than a predefined threshold, one sensor node will go through a fast parent node reselection phase before disseminating code packets to its child nodes. Take Figures 7 and 8 as an example: after the first round of retransmission, node E will use (12) to estimate link correlation. The value of s in Figure 7 is $(1 - 1/5) = 80\%$, while that in Figure 8 is $(1 - 4/9) = 56\%$. Therefore, it is more likely for node E in Figure 8 to go through a reselection phase than node E in Figure 7.

Obviously, this phase will only prolong the dissemination completion time if link correlation remains stable during codes dissemination. However, this phase will improve the performance when the environment is highly dynamic. Therefore, it is necessary to achieve a balance between the improvement brought by this phase and extra delay results from this phase. Detailed analysis on delay is shown in Section 5.2 and its impact in our experiment is shown in Section 6.4.1.

4.3.5. The Delay-Aware Program Dissemination Algorithm. See Scheme 1.

5. Analysis on Delay

5.1. Analysis on Delay of Codes Dissemination. First, we analyze transmission delay for one node to broadcast N code packets to its child nodes. Changes on link correlation

```

Initialize: Each sensor node obtains its hop count through a flooding originated from the base station;
Initial parent node selection phase
(1) For each node  $i$  Do
(2)   Broadcast HELLO messages;
(3)   For each node  $j$  who receives HELLO messages above Do
(4)     Reply ACKs to  $i$ ;
(5)   End for
(6)   Construct reception vectors for each node  $j$ ;
(7)   Broadcast reception vectors;
(8)   For each node  $k$  who has the highest correlation with node  $i$ 
(9)     Send a CHOSEN message to node  $i$ ;
(10)  End for
(11) End for
Data dissemination phase
(1) For  $h = 0$  to  $X - 1$  ( $h$ : hop count;  $X$ : the largest hop count in the network)
(2)   Use equation (12) to estimate link correlation;
(3)   If  $s >$  Threshold
(4)     For each node  $i$  whose hop count is  $h$  Do
(5)       Broadcast code packets;
(6)       For each node  $j$  who receives packets above and its hop count equals  $h + 1$  Do
(7)         If the parent node of  $j$  is  $i$ 
(8)           Reply ACKs to  $i$ ;
(9)         Else
(10)          Reply ACKs to its own parent;
(11)        End for
(12)      End for
(13)    Else
(14)      Go through the next phase and then return Step (4);
(15)    End for
Fast parent node reselection phase
(1) For each node  $i$  whose hop count is  $h$  Do
(2)   Broadcast HELLO messages;
(3)   For each node  $j$  who receives HELLO messages above Do
(4)     if hop count of  $j$  is  $h + 1$ 
(5)       Reply ACKs to  $i$ ;
(6)     End for
(7–11) Same to the first phase;

```

SCHEME 1: Delay-aware program dissemination scheme.

during codes dissemination are ignored here and necessary notations are given as follows:

i : parent node

X : number of child nodes

n_k : child node ID ($k = 1, 2, \dots, X$)

q_i : link quality when i transmits packets

q_{n_k} : link quality when k replies ACKs

N : number of code packets that i transmits

c_{n_k} : link correlation between n_k and i

According to (3), we can obtain number of code packets that are successfully received by all child nodes at the first transmission.

$$\sum_{m=1}^N V_{n_k}(m) = Nq_{n_k}, \quad (13)$$

$$\sum_{m=1}^N (V_{n_1}(m) \times V_{n_2}(m) \times \dots \times V_{n_X}(m)) = Nc_{n_k}q_{n_k}, \quad (14)$$

where $V_{n_k}(m)$ is the reception on the m th packet of child node n_k . The left part of (13) indicates number of packets that need not to be retransmitted.

According to (14), we also have

$$c_{n_1}q_{n_1} = c_{n_2}q_{n_2} = \dots = c_{n_X}q_{n_X}. \quad (15)$$

Therefore, the value of k is irrelevant to number of packets that are successfully received by all child nodes in (13), which is a variable shared by all child nodes.

At the first time, i needs to transmit all N code packets.

At the second time, the number is

$$N - Nc_{n_k}q_{n_k}. \quad (16)$$

And, at the third time, the number is

$$\begin{aligned}
 & N - Nc_{n_k}q_{n_k} - (N - Nc_{n_k}q_{n_k}) \times c_{n_k}q_{n_k} \\
 &= N - 2Nc_{n_k}q_{n_k} + N(c_{n_k}q_{n_k})^2 \\
 &= N - 2Nc_{n_k}q_{n_k} + o(c_{n_k}q_{n_k}) \approx N - 2Nc_{n_k}q_{n_k}.
 \end{aligned} \tag{17}$$

Besides, the expected number of transmissions T is $1/\min(q_i q_{n_k})$. Hence, total number of code packets that parent node i needs to transmit can be obtained through the following equation:

$$D = \sum_{m=1}^T (N - N(m-1)c_{n_k}q_{n_k}). \tag{18}$$

However, calculated number of packets that need to be transmitted above can be imprecise in the real world, since link correlation will change during codes dissemination and link quality will not always be the same to the link quality estimated by (1).

After obtaining D , transmission delay can also be calculated. A complete process of transmission is described as follows. First, parent node i broadcasts N code packets continuously and starts to receive ACKs from its child nodes. Upon receiving some or all of these packets, one child node will reply with ACKs that contain information on which packets are successfully received. After keeping receiving ACKs for a long time which ensures that all potential ACKs can arrive at the parent node, i will broadcast left packets which are not successfully received by all child nodes. The process described above will cycle until all its child nodes receive N code packets; then parent node i will start to broadcast the next N packets. Figure 14 is a sequence diagram of transmission.

t_{DATA} is time spent on transmitting code packets, t_{ACK} is time spent on replying ACKs, and τ is an extra time to ensure that all potential ACKs can arrive. Hence, the delay for one node to broadcast N code packets to its child nodes is

$$t = \sum_{m=1}^T \left(\frac{D_m \times S_{\text{DATA}}}{R_D} + \frac{D_m \times S_{\text{ACK}}}{R_D} + \tau \right), \tag{19}$$

where D_m is number of code packets that need to be transmitted at the m th time, S_{DATA} and S_{ACK} are size of code packet and ACK separately, and R_D is data transmission rate. Figure 15 shows transmission delay for one node to transmit ten code packets to all its one-hop downstream nodes in several cases where link correlation between these nodes is set to 20%, 40%, 60%, and 80% manually. Transmitting power of nodes in Figure 15 is same to the green curve in Figure 12, while S_{DATA} , S_{ACK} , and R_D are same to Section 6.1. With the distance from the base station increasing, gaps between transmission delay start to shrink, since a more reliable link tend to weaken benefits brought by link correlation.

5.2. Analysis on Delay of Parent Node Reselection. Due to the fact that the environment of wireless sensor network

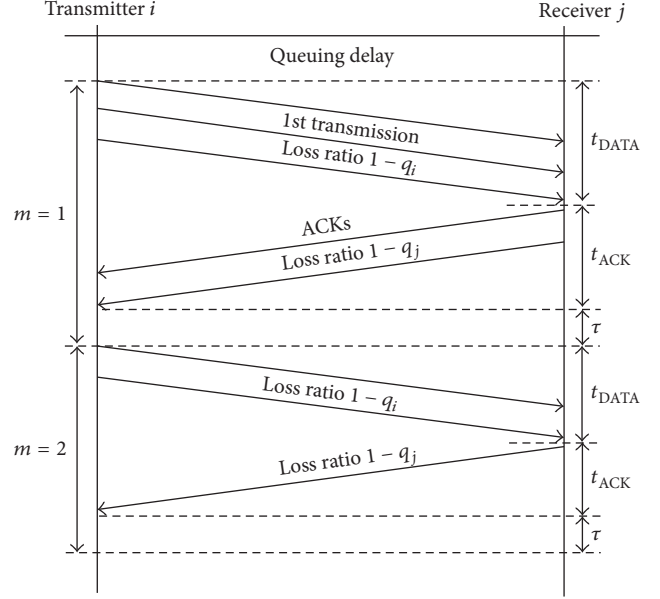


FIGURE 14: Sequence diagram of transmission.

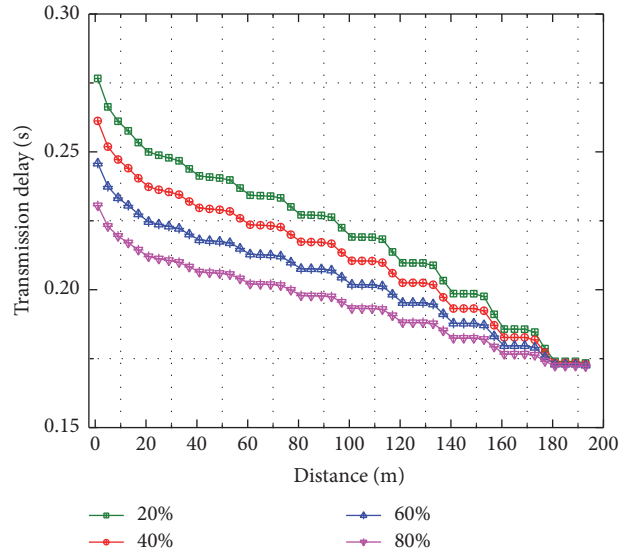


FIGURE 15: Transmission delay with different link correlation.

can be highly dynamic, link correlation obtained in initial parent node selection phase tends to change during codes dissemination phase. An example in Section 4.1.2 shows that the improvement brought by adopting link correlation can be greater after reselecting parent nodes during codes dissemination. However, this new parent node selection phase will also bring extra delay to codes dissemination. Therefore, the delay brought by fast parent node reselection phase is analyzed in this section.

Data transmission of this phase is shown in Figure 13. In this phase, reliabilities of HELLO messages and ACKs are not guaranteed, while that of V (packets contain information on reception vectors) and CHOSEN message are assured through timeout retransmission mechanism.

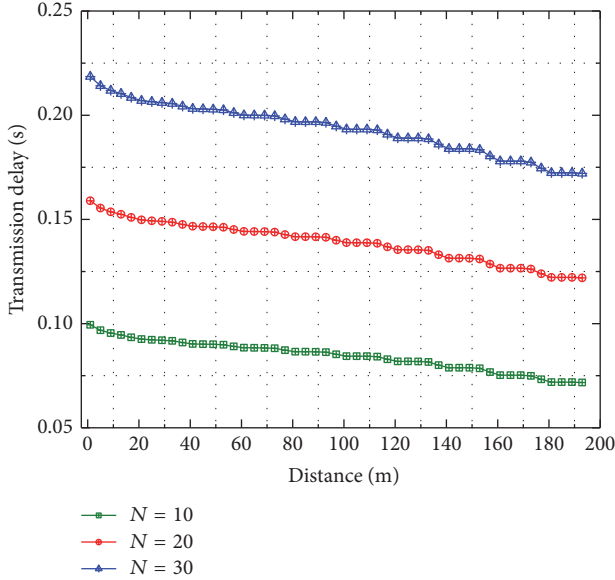


FIGURE 16: Expected delay brought by fast parent node selection phase.

First, sensor nodes continuously broadcast N HELLO messages to one-hop downstream sensors in their broadcast domain. Second, they start to receive ACKs sent by one-hop downstream nodes. An extra time τ is also added here to ensure the arrival of ACKs. Then, they construct reception vectors according to these ACKs and broadcast V , which is composed of packets that contain information on reception vectors. After receiving V , one-hop downstream nodes will calculate link correlations for each node that broadcasts V and select the node with which they have highest link correlation as their parent node. At last, a CHOSEN message is sent to inform this node that it has been chosen.

According to the description above, the expected delay brought by fast parent node reselection phase is shown in the following equation:

$$t_d = \frac{D_{\text{HELLO}} + D_{\text{ACK}}}{R_D} + \tau + T \times \left(\frac{D_V + D_{\text{CHOSEN}} + 2D_{\text{ACK}}}{R_D} + 2\tau \right), \quad (20)$$

where D_* is data size on corresponding packet and T is the expected number of transmissions. Figure 16 shows transmission delay for nodes to go through fast parent node reselection phase with N HELLO messages; additional delay brought by the phase reduces with distance from the base station increasing since a higher transmitting power also benefits this phase.

6. Experimental Evaluation

In this section, a simulation experiment is given to evaluate delay-aware program dissemination (DAPD). First, parameters of network and sensor nodes are introduced. Second, we conduct the experiment according to parameters above

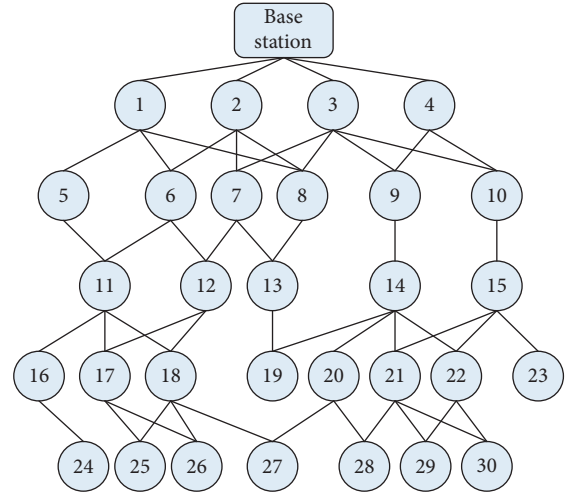


FIGURE 17: The topology of our network.

TABLE 1: Parameter setting on sensors.

Parameter	Value
Working voltage	2.7–3.3 (V)
Transmitting power	–20–10 (dBm)
Receive sensitivity	–101 (dBm)
Transmission rate	<76.8 (kbps)

and experiment results are compared with deluge and Link-Correlation-Aware Data Dissemination (CD). Third, we analyze the impact of changes on network parameters.

6.1. Parameters Setting. The wireless sensor network is composed of one base station and thirty sensor nodes. Each node can have 1 to 4 links (represented by black line) with one-hop downstream nodes and the largest hop count in the network is set to 5. The topology of network is generated randomly with parameters above and final result is shown in Figure 17.

The energy of base station is considered to be infinite; therefore its transmitting power is high enough. Initial energy of sensors in the network is 0.5 J and transmission radius of all sensors is 20 m. Total data size on codes is 40 kB, while data sizes on each code packet S_{DATA} and ACK S_{ACK} are 20 bytes and 5 bytes separately. Besides, τ is designated as 5 (ms). Parameter settings on sensors are shown in Table 1.

To ensure that all sensor nodes can be updated successfully through codes, each sensor will broadcast 16 packets continuously and start to receive ACKs after then. Following packets will not be transmitted until it receives all its child nodes' ACKs for these 16 packets. Besides, change on link correlation during codes dissemination is ignored here and analyzed independently in Section 6.4.1.

The transmitting power of deluge and CD is shown in Table 2 and that of DAPD after enhancing without reducing the lifetime of network is shown in Table 3. Before reprogramming, they all have already collected data packets for 100 times and transmitted them back to the base station with $P_t = 0$ (dBm).

TABLE 2: Transmitting power of deluge and CD.

h	1	2	3	4	5
P_t (dBm)	0	0	0	0	0
q	0.6710	0.6710	0.6710	0.6710	0.6710

TABLE 3: Transmitting power of DAPD.

h	1	2	3	4	5
P_t (dBm)	0	0.1	0.8	1.8	3
q	0.6710	0.7109	0.8078	0.9069	0.9625

h is hop count, P_t is transmitting power, and q is theoretical link quality calculated through (1).

6.2. Selected Parent Nodes. First, each sensor node will select its parent node during initial parent node selection phase described in Section 4.3.2.

For example, reception vectors kept by nodes with hop count 1 are shown in Table 4.

P is the ID of transmitter, N is the ID of receiver, and $V(i)$ is the i th element in the reception vector (1 indicates a successfully received HELLO message, while 0 indicates a failure). Figure 18 shows selected parent nodes after this phase. Top of red lines denotes selected parents of nodes. Grey lines indicate that although those one-hop upstream sensors are not parent nodes, code packets broadcast by them are still possible to be received by nodes connected by the bottom of grey lines.

6.3. Comparison with Deluge and CD

6.3.1. Evaluation on Delay. Average transmission delay of deluge, CD, and DAPD is shown in Figure 19. i in x -axis indicates codes dissemination from sensors with hop count $i-1$ to sensors with hop count i . According to Figure 19, average transmission delay of DAPD is 19.05% and 16.65% smaller than deluge and CD separately. Two reasons can account for this improvement: (1) DAPD adopts link correlation compared with deluge; (2) DAPD intelligently enhances transmitting power to reduce delay further compared with CD. Figure 20 presents the distribution of link correlation and link quality collected from all reception vectors, from which we can observe that although link correlation does not present any regular distribution, link quality in DAPD is higher than CD appreciably due to the enhancement of transmitting power.

6.3.2. Evaluation on Energy Consumption. Two main sources of energy consumption come from transmitting and receiving code packets, ACKs. Therefore, we make comparisons on the sum of these two sources. Results are shown in Figure 21. Due to the fact that there is no need for sensors with hop count 5 to transmit code packets, their energy consumption is far smaller than upstream nodes and is

therefore ignored. From Figure 21, we can see that on one hand energy consumption of DAPD will become larger with hop count increasing, which meets our scheme that utilizes excess energy in far nodes to enhance transmitting power; on the other hand, the energy consumption of both CD and DAPD is smaller than deluge on sensors with hop count smaller than 4, where link correlation plays a more important role. Therefore, the adoption of link correlation can reduce delay and energy consumption simultaneously. Besides, Figure 22 shows the distribution of residual energy after disseminating codes for one and three times; DAPD slightly relieves the unbalanced residual energy distribution of sensors.

6.4. Impact of Network Parameters

6.4.1. The Volatility of Environment. When the environment of wireless sensor network is highly dynamic, parent nodes selected during initial parent node selection phase may be outdated and unable to make full use of link correlation during codes dissemination phase. For example, Figure 23 shows a possible situation where average link correlations between sensors with hop count 3 and 4 and sensors with hop count 4 and 5 are modified from 62.47% and 74.63% to 20% manually. Average transmission delay of CD and DAPD increases toward deluge after changing link correlations manually.

Through using (12) to estimate link correlation, sensor nodes with hop count 3 and 4 will make the assumption that it is necessary to go through a fast parent node reselection phase. Data sizes on HELLO message, ACK, and CHOSEN message are all 5 bytes, while that on packet of V is same as S_{DATA} for convenience. Compared with time used for transmission, times spent on constructing reception vectors and calculating link correlation are far smaller; therefore they are ignored when measuring delay. Figure 24 shows the delay before and after adopting fast parent node reselection phase, from which we can see that although transmission delay cannot recover to previous unchanged level, it improves a lot compared with taking no action when link correlation changes. Besides, transmitting power of sensors becomes higher with distance from base station increasing, which leads to a lower delay, and this improvement also benefits the fast parent node reselection phase. Therefore, the delay brought by this phase will decrease when it happens on sensors with a large hop count.

6.4.2. Data Size and Packet Size. Figure 25 shows dissemination completion time (DCT) of codes dissemination with different packet size, while Figure 26 shows dissemination completion time (DCT) with different total size of data.

6.4.3. Number of Hop Counts and Nodes. The scale of wireless sensor network mainly depends on the number of hop counts and nodes, which tend to have great impacts on the delay. On one hand, the increase on hop count will directly increase number of times that sensor nodes transmit and retransmit packets. On the other hand, amount of data that

TABLE 4: Reception vectors kept by nodes.

P	N	$V(1)$	$V(2)$	$V(3)$	$V(4)$	$V(5)$	$V(6)$	$V(7)$	$V(8)$	$V(9)$	$V(10)$
①	⑤	1	1	0	1	1	0	0	1	1	1
①	⑥	0	1	1	1	1	0	1	1	1	1
①	⑧	1	0	1	1	1	1	1	1	1	0
②	⑥	1	0	0	0	0	1	1	0	1	0
②	⑦	1	1	1	1	0	1	1	1	1	1
②	⑧	0	1	1	0	0	1	1	0	1	0
③	⑦	1	0	1	1	0	0	1	0	0	0
③	⑧	1	0	1	0	1	1	0	0	1	1
③	⑨	1	1	1	1	1	0	1	1	1	1
③	⑩	0	0	1	1	1	0	1	1	0	1
④	⑨	0	1	0	1	0	1	0	1	1	1
④	⑩	1	0	0	0	0	1	1	1	0	1

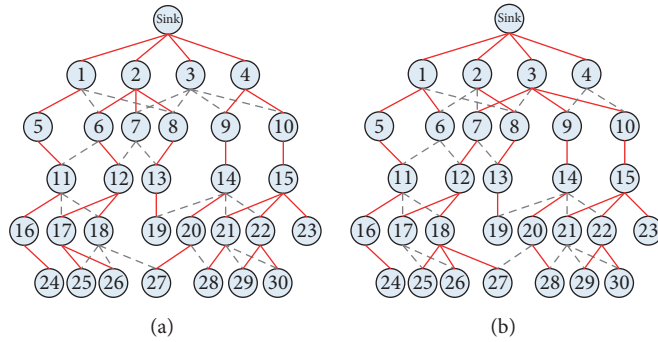


FIGURE 18: Selected parent nodes in CD and DAPD. (CD: (a); DAPD: (b)).

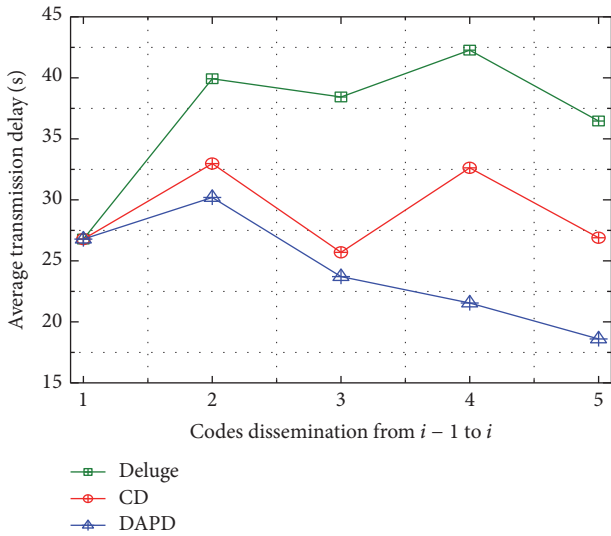


FIGURE 19: Average transmission delay.

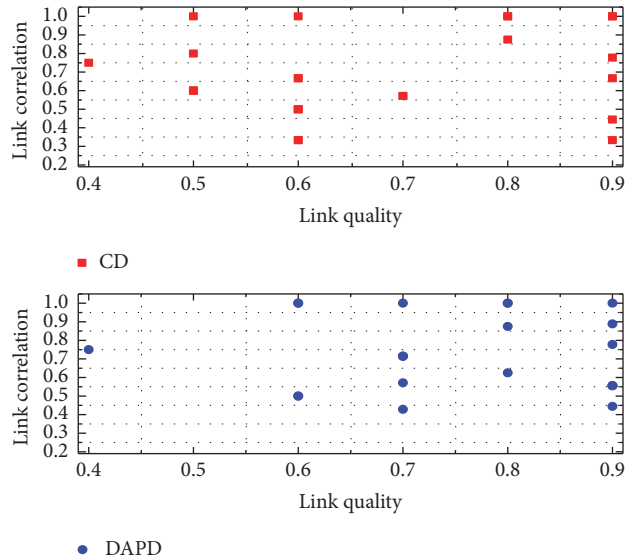


FIGURE 20: Distribution of link quality and link correlation.

generates during data collection phase will also increase with number of nodes becoming larger, which leads to more energy consumption and a more unbalanced residual energy

distribution. Therefore, it is meaningful to analyze the impact of network's scale on our scheme. Since the distribution on link correlation is irregular according to Figure 20, the impact

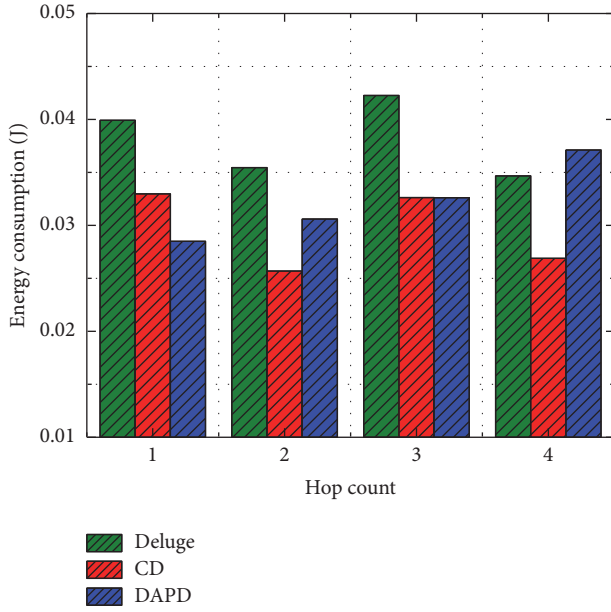


FIGURE 21: Energy consumption of deluge, CD, and DAPD.

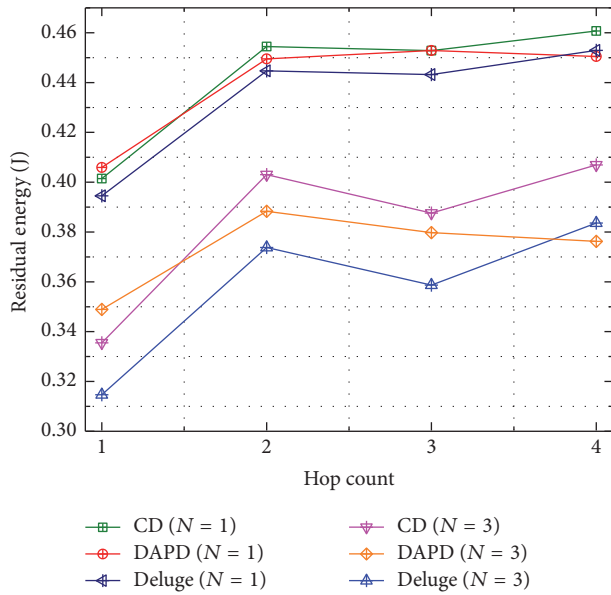


FIGURE 22: Distribution of residual energy.

concentrates on the enhancement of transmitting power. First, Figure 27 shows data load on nodes with different distance from base station during data collection phase (ρ_0 is the original density of nodes); gaps between nodes around and nodes far from the base station become wider with number of hop count and nodes increasing.

With a different distribution of data load after the scale of network changing, transmitting power during codes dissemination phase will also be different from Table 3, which is shown in Figure 28.

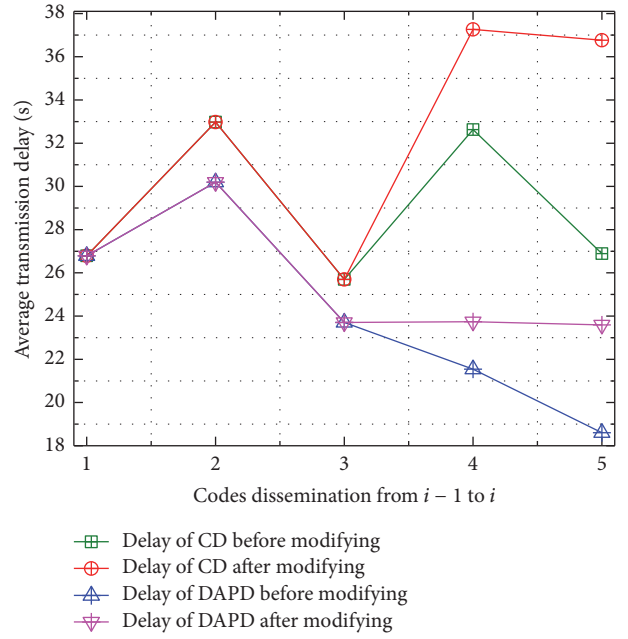


FIGURE 23: Delay of CD and DAPD after modifying link correlation manually.

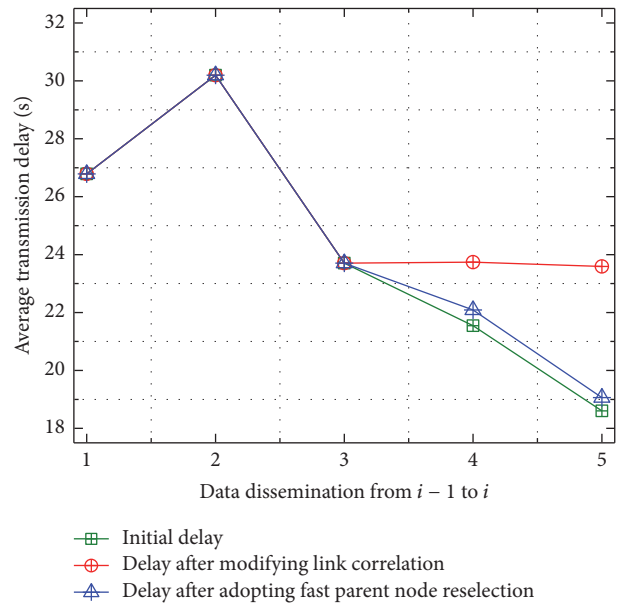


FIGURE 24: Delay for DAPD after adopting fast parent node reselection phase.

6.4.4. *The Period of Codes Dissemination.* Apart from hop count and number of nodes, frequency of codes dissemination will also influence our scheme. Generally, the enhancement of transmitting power will be smaller with a higher frequency. The period T is defined as follows:

$$T = \frac{n_c}{n_d}, \quad (21)$$

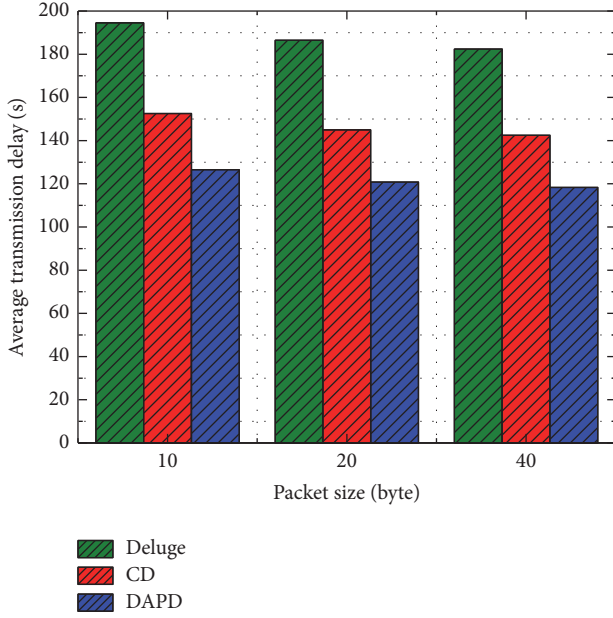


FIGURE 25: DCT with different packet size.

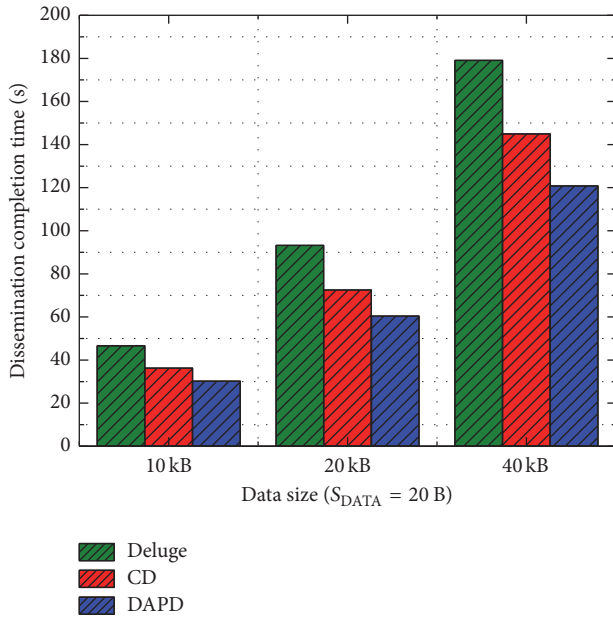


FIGURE 26: DCT with different total size of data.

where n_c is times of collecting data instead of times of sending back data packets because the latter one is influenced by link quality and can be unpersuasive when measuring period and n_d is times of codes dissemination. Impacts of different periods on the enhancement of transmitting power are shown in Figure 29.

7. Conclusion

In this paper, we propose a novel codes dissemination scheme which focuses on reducing delay and being adaptive to highly

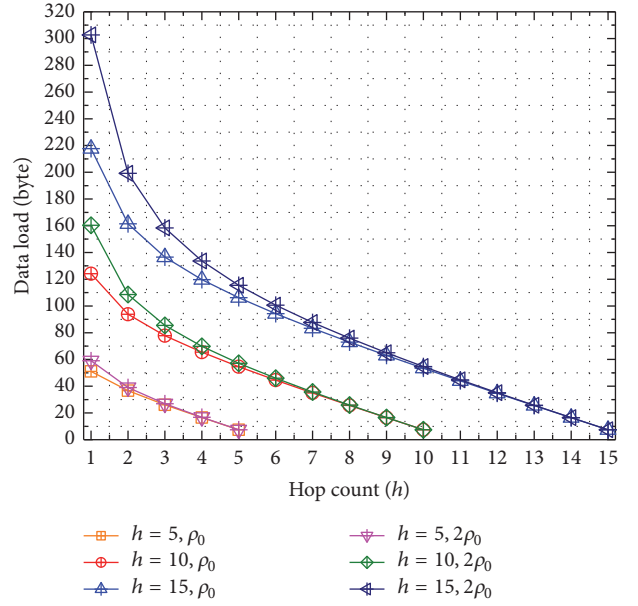


FIGURE 27: Data load on sensors in wireless sensor networks with different scales.

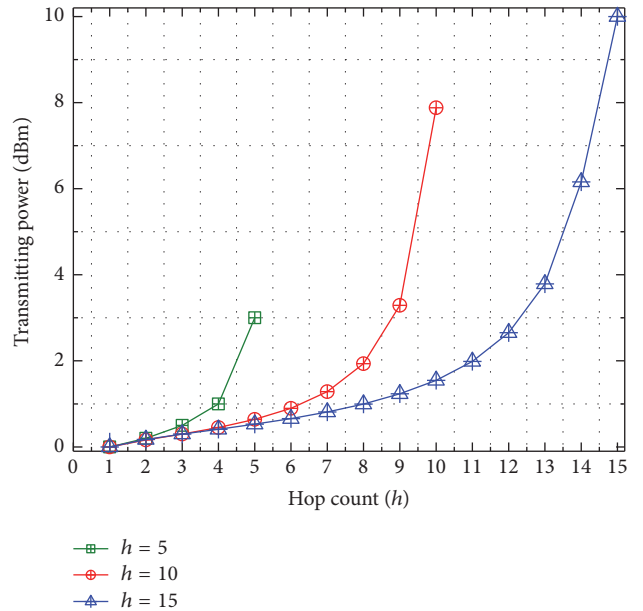


FIGURE 28: The enhancement of transmitting power in networks with different scales.

dynamic environment. In general, it has three salient features: (1) utilize link correlation in wireless sensor network to reduce number of transmissions, (2) enhance transmitting power during codes dissemination to improve link quality, and (3) go through a fast parent node reselection phase when the environment changes in order to reuse link correlation. A simulation experiment shows that, compared with deluge and another codes dissemination scheme CD, our scheme achieves better performance in both static and dynamic environments.

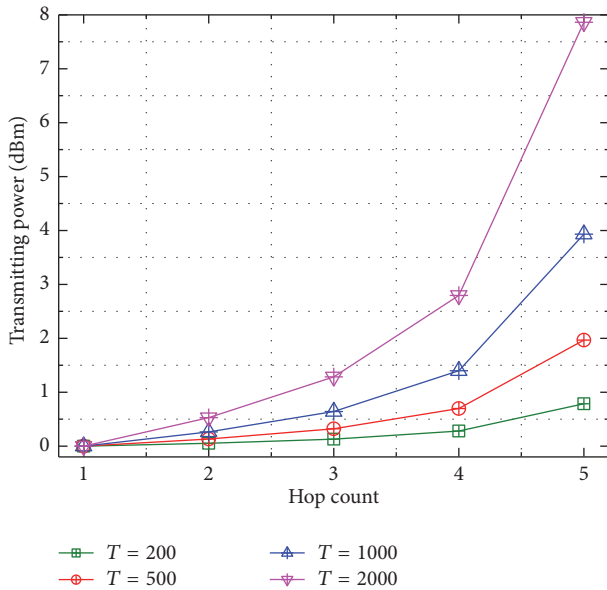


FIGURE 29: The enhancement of transmitting power with different period T .

However, as mentioned in Section 6.3.2, one disadvantage on DAPD is that residual energy in sensor nodes with largest hop count is not exploited, since there is no need for these nodes to broadcast code packets during codes dissemination phase, which is an important source of energy consumption for other nodes in the network.

Our future work includes integrating code packets transmission between sensor nodes with same hop count into DAPD to improve the performance further and a comprehensive research on the relation between link quality and link correlation.

Competing Interests

The authors declare that there are no competing interests regarding the publication of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61379110, 61370229, and 61305144), the National Basic Research Program of China (973 Program) (2014CB046305), the National Key Technology R&D Program of China (2014BAH28F02), and the S&T Projects of Guangdong Province (2014B010103004, 2014B010117007, 2015A030401087, 2015B010110002, 2015B010109003, and 2016B010109008).

References

- [1] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proceedings of the IEEE 29th International Conference on Advanced Information Networking and Applications (AINA '15)*, pp. 687–694, IEEE, Gwangju, South Korea, March 2015.
- [2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): a vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [3] CISCO's Technology News Site, <https://newsroom.cisco.com/ioe>.
- [4] M. Stauffer, "Connecting the internet of everything," in *Proceedings of the Hot Chips 26 Symposium (HCS '14)*, vol. 26, pp. 38–10, IEEE, Cupertino, Calif, USA, August 2014.
- [5] S. He, J. Chen, X. Li et al., "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1268–1282, 2014.
- [6] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: when QoE meets QoP," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 74–80, 2015.
- [7] B. Fan-Yu, X. Wang, and Q. Zhang, *Data Mining Model Based on Cloud Computing in the Internet of Things*, Computer & Information Technology, 2012.
- [8] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 3, pp. 312–325, 2016.
- [9] K. Suto, H. Nishiyama, N. Kato, and CW. Huang, "An energy-efficient and delay-aware wireless computing system for industrial wireless sensor networks," *IEEE Access*, vol. 3, pp. 1026–1035, 2015.
- [10] Z. Tang, A. Liu, and C. Huang, "Social-aware data collection scheme through opportunistic communication in vehicular mobile networks," *Included in Special Section in IEEE Access: Recent Advances in Socially-aware Mobile Networking*, vol. 4, pp. 6480–6502, 2016.
- [11] X. Liu, T. Wei, and A. Liu, "Fast program codes dissemination for smart wireless software defined networks," *Scientific Programming*, vol. 2016, Article ID 6907231, 21 pages, 2016.
- [12] D. Zeng, P. Li, S. Guo, T. Miyazaki, J. Hu, and Y. Xiang, "Energy minimization in multi-task software-defined sensor networks," *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3128–3139, 2015.
- [13] V. Sharma, I. You, and R. Kumar, "Energy efficient data dissemination in Multi-UAV coordinated wireless sensor networks," *Mobile Information Systems*, vol. 2016, Article ID 8475820, 13 pages, 2016.
- [14] W. Dong, C. Chen, X. Liu, G. Teng, J. Bu, and Y. Liu, "Bulk data dissemination in wireless sensor networks: modeling and analysis," *Computer Networks*, vol. 56, no. 11, pp. 2664–2676, 2012.
- [15] R. Adline Freeda and R. N. Sharmila, "A review of bulk data dissemination protocols for reprogramming in WSN," in *Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES '16)*, pp. 1–4, IEEE, Chennai, India, February 2016.
- [16] Q. Wang, Y. Zhu, and L. Cheng, "Reprogramming wireless sensor networks: challenges and approaches," *IEEE Network*, vol. 20, no. 3, pp. 48–55, 2006.
- [17] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 81–94, November 2004.

- [18] H. Lim and C. Kim, "Flooding in wireless ad hoc networks," *Computer Communications*, vol. 24, no. 3-4, pp. 353-363, 2001.
- [19] Z. Zhao, W. Dong, J. Bu, Y. Gu, and C. Chen, "Link-correlation-aware data dissemination in wireless sensor networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 9, pp. 5747-5757, 2015.
- [20] X. Zheng, J. Wang, W. Dong, Y. He, and Y. Liu, "Bulk data dissemination in wireless sensor networks: analysis, implications and improvement," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1428-1439, 2016.
- [21] A. Hagedorn, D. Starobinski et al., "Rateless deluge: over-the-air programming of wireless sensor networks," in *Proceedings of the IEEE International Conference on Information Processing in Sensor Networks (IPSN '08)*, Proceeding of ACM, April 2008.
- [22] Y. Hu, M. Dong, K. Ota, A. Liu, and M. Guo, "Mobile target detection in wireless sensor networks with adjustable sensing frequency," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1160-1171, 2016.
- [23] X. Zheng, J. Wang, W. Dong, Y. He, and Y. Liu, "Survival of the fittest: data dissemination with selective negotiation in wireless sensor networks," in *Proceedings of the 10th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '13)*, pp. 443-451, IEEE, Hangzhou, China, October 2013.
- [24] Q. Yang, S. He, J. Li et al., "Energy-efficient probabilistic area coverage in wireless sensor," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 367-377, 2015.
- [25] Y. Liu, A. Liu, and Y. Hu, "FFSC: an energy efficiency communications approach for delay minimizing in internet of things," *IEEE Access*, vol. 4, pp. 3775-3793, 2016.
- [26] S. S. Kulkarni and L. Wang, "MNP: multihop network reprogramming service for sensor networks," in *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems*, June 2005.
- [27] R. Panta, S. Bagchi, I. Khalil et al., "Single versus multi-hop wireless reprogramming in sensor networks," in *Proceedings of the 4th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM '08)*, Innsbruck, Austria, 2008.
- [28] Q. Zhang and A. Liu, "An unequal redundancy level-based mechanism for reliable data collection in wireless sensor networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, article 258, 2016.
- [29] H. Cheng, N. Xiong, X. Huang, and L. T. Yang, "An efficient scheduling model for broadcasting in wireless sensor networks," in *Proceedings of the IEEE 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum (IPDPSW '13)*, pp. 1417-1422, May 2013.
- [30] M. Zuniga and B. Krishnamachari, "Analyzing the transitional region in low power wireless links," in *Proceedings of the 14th Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pp. 517-526, 2004.
- [31] Z. Tang, A. Liu, Z. Li, Y.-J. Choi, H. Sekiya, and J. Li, "A trust-based model for security cooperating in vehicular cloud computing," *Mobile Information Systems*, vol. 2016, Article ID 9083608, 22 pages, 2016.
- [32] R. Xie, A. Liu, and J. Gao, "A residual energy aware schedule scheme for WSNs employing adjustable awake/sleep duty cycle," *Wireless Personal Communications*, vol. 90, no. 4, pp. 1859-1887, 2016.
- [33] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, USA, 2001.
- [34] M. Perillo, Z. Cheng, and W. Heinzelman, "On the problem of unbalanced load distribution in wireless sensor networks," in *Proceedings of the IEEE Global Telecommunications Conference Workshops (GLOBECOM '04)*, pp. 74-79, December 2004.
- [35] L. T. Yang, A. B. Waluyo, J. Ma, L. Tan, and B. Srinivasan, *Energy-Efficient Pattern Recognition for Wireless Sensor Networks*, John Wiley & Sons, Hoboken, NJ, USA, 2010.
- [36] S. He, J. Chen, and F. Jiang, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1931-1942, 2013.
- [37] Y. Liu, A. Liu, and Z. Chen, "Analysis and improvement of send-and-wait automatic repeat-request protocols for wireless sensor networks," *Wireless Personal Communications*, vol. 81, no. 3, pp. 923-959, 2015.

