A Hybrid Reliable Routing Algorithm Based on LQI and PRR in Industrial Wireless Networks

Jie Li, Yang Pan, Shijian Ni, and Feng Wang

Northeastern University, Shenyang, China

Correspondence should be addressed to Yang Pan; panyang@stumail.neu.edu.cn

Received 28 April 2021; Revised 18 July 2021; Accepted 15 August 2021; Published 6 September 2021

Academic Editor: Hui Cheng

Copyright © 2021 Jie Li 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.

In Industrial Wireless Networks (IWNs), the communication through Machine-to-Machine (M2M) is often affected by the noise in the industrial environment, which leads to the decline of communication reliability. In this paper, we investigate how to improve route stability through M2M in an industrial environment. We first compare different link quality estimations, such as Signal-Noise Ratio (SNR), Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI), Packet Reception Ratio (PRR), and Expected Transmission Count (ETX). We then propose a link quality estimation combining LQI and PRR. Finally, we propose a Hybrid Link Quality Estimation-Based Reliable Routing (HLQEBRR) algorithm for IWNs, with the object of maximizing link stability. In addition, HLQEBRR provides a recovery mechanism to detect node failure, which improves the speed and accuracy of node recovery. OMNeT++-based simulation results demonstrate that our HLQEBRR algorithm significantly outperforms the Collection Tree Protocol (CTP) algorithm in terms of end-to-end transmission delay and packet loss ratio, and the HLQEBRR algorithm achieves higher reliability at a small additional cost.

1. Introduction

In Industrial Wireless Network (IWN) communication, wireless network has been popularized as a flexible substitute for wired network and has gradually become a research hot spot on the field of industrial networks [1]. Compared with wired networks, IWNs are easier to maintain on-site equipment and have the advantages of easy and quick installation. However, IWNs still have shortcomings: first, because these networks transmit data wirelessly, the delay is higher than that of wired networks; secondly, there are a large number of malicious attacks in wireless transmission in industrial environments, which will lead to the unsafe transmission of data by wireless network links; finally, in the existing industrial environment, most industrial devices in wireless networks still require battery power, so once the power of the device is exhausted, it means that the device will fail. In industrial applications, the data being transmitted in wireless networks may play a vital role in production safety, so the data should be transmitted to the receiving end reliably and as soon as possible. To improve reliability, WirelessHART adopts TDMA and Automatic Repeat Request (ARQ) techniques [2] but leaves the details and implementation of most scheduling algorithms to the vendor. In order to ensure that the information is not affected by background interference, path loss, multipath fading, and node failure in the transmission process, a variety of reliability enhancement techniques are proposed when discussing the challenges of wireless network security and reliability [3]. Redundancy is a way to improve reliability [4], which can be implemented on different layers and appear in the form of redundant packet content and error correction code (physical layer), repeated transmission (MAC layer), installing relay, or concurrent transmission with multiple paths. Hong et al. proposed an object-oriented routing algorithm to realize multipath diversity for a time-invariant network environment [5]. Although the above method [2–5] can improve the transmission reliability of IWNs, the cost is to consume more energy. The signal of the wireless communication node is sent out from the transmitting end, and it has experienced large-scale fading and small-scale fading [6]. When the signal reaches the receiving end, the strength of the signal will be significantly reduced. Moreover, the noise interference of the industrial field environment makes the detection and demodulation
of the signal by the receiving end more difficult. In addition, the wireless link is asymmetric, and the stability between the two nodes is asymmetric. Therefore, the unreasonable link quality estimation will lead to the routing performance which can not reach the expected level even considering the reliability and other aspects of the routing protocol. Accuracy, reactivity, and stability should be considered in link quality assessment. The hardware-based link quality estimation can be read directly by the wireless receiving end. Its advantage is that it does not require additional computation, but its limitation is that it can only be obtained by the received packet. Therefore, when the wireless link loses a large number of packets, hardware-based link quality estimates tend to over-estimate the link quality. Although hardware-based link quality estimation can be used to determine whether a link is in particularly good or bad condition without additional overhead, it does not provide fine-grained link quality estimation [7]. The feature of software-based link quality estimation is that the link quality is not directly estimated by hardware reading but by calculation. This method may require a large number of probe packages, and the time to send and receive the probe package is relatively long, which increases the time cost and energy consumption. Relatively speaking, it can provide fine-grained link quality estimation.

In addition, the routing algorithm used in the network will also affect the estimation of link quality. First of all, the adoption of non-QoS routing protocols that do not support reliability or do not involve link quality in the calculation process may lead to poor routing reliability and high packet loss rate. Secondly, different routing algorithms have their own emphases. Even if some aspects are considered, others may be ignored, and sometimes, the ideal optimal solution cannot be achieved or simply does not exist. There are also problems in the design of the algorithm itself, such as unreasonable logical judgment or defects in the calculation process, which may lead to insufficient utilization of information and data. It is also possible that the complexity of the algorithm is not suitable for IWNs. Finally, excessive pursuit of reliability for routing algorithms can lead to other performance degradation, such as increasing end-to-end delay and energy consumption; such routing algorithms are also inappropriate.

The application of IWNs requires high reliability, so some measures must be taken to deal with the node failure. Industrial environments can be very complex, and node failures may have different possibilities. The nodes themselves may fail due to energy exhaustion or may be destroyed due to physical damage. At the same time, the node may not fail but the transmission failure due to poor link quality is mistaken as that the node has failed and enters the node failure processing, which reduces the network performance. On the one hand, it is necessary to design an effective node failure treatment mechanism; on the other hand, we should pay attention to the speed of failure detection and the precision of accuracy and logical judgment.

In this paper, a reliable routing algorithm based on the most reliable routing (HLQEBRR) is proposed to ensure that the nodes in the network can obtain sufficient reliability assurance:

1. We analyze the link quality estimation from both hardware and software and adopt a link quality estimation that is more suitable for IWNs, so as to improve the reliability of link communication.
2. We propose a hybrid reliable routing algorithm based on link quality estimation to guarantee the reliability in IWNs.
3. We propose a necessary collection mechanism to collect information in the network and a coping strategy when the nodes run out of energy or fail due to other accidents.

Experimental simulation shows that the HLQEBRR algorithm is 5% higher than the CTP (Collection Tree Protocol) in the average end-to-end delivery rate and 21.2% in reliability. The remainder of this paper is organized as follows. Related work is described in Section 2 to illustrate link quality estimation and existing routing protocols. Section 3 presents the network model and our routing algorithm. Simulation and results are presented in Section 4. Finally, Section 5 concludes this paper.

2. Related Work

2.1. Link Quality Estimation. Estimation of link quality can be divided into hardware-based and software-based. The hardware-based estimation usually takes the Signal-Noise Ratio (SNR), Received Signal Strength Indicator (RSSI), and Link Quality Indicator (LQI) as reference, while the software-based estimation takes the Packet Reception Ratio (PRR) and Expected Transmission Count (ETX) for reference [7]. Four types of link quality estimation are briefly described in the following. Received Signal Strength Indicator (RSSI) can quickly and accurately estimate link quality. Srinivasan et al. concluded through experiments that if the RSSI value is greater than 87 dBm, the packet reception rate can reach 99%. When below this value, the RSSI reduction of 2 dBm, link state changes dramatically [8]. The value of the RSSI has very good stability, reflected in its standard deviation less than 1 dBm, so a single RSSI reading can be used to determine whether the link quality is within an acceptable range. Since the noise substrates of different nodes are different, it is a better choice to estimate link quality by using Signal-Noise Ratio (SNR) than the RSSI method of adding pure received signal and receiving end noise substrate. The Link Quality Indicator (LQI) can immediately determine the state of the link as RSSI. When the LQI is close to 110, the packet receiving rate is close to 100, and the variance is very low. So a single LQI reading can determine the link quality state. When the link is in the middle quality, the variance of the LQI will increase, and the reading of a single LQI cannot meet the requirement of accurately estimating the link quality. Therefore, it is necessary to obtain a large number of LQI and obtain the mean value to provide an accurate link quality estimation. Boano and others suggest that LQI variance can be used to distinguish link quality [9]. The reason why support LQI is better than RSSI is that it is more relevant to PRR than the average LQI. Package
Receiving Rate (PRR) is a receiving end estimation, which is widely used in routing protocols and can be used as an unbiased measure to evaluate the accuracy of hardware link estimation. If a hardware-based link estimation is correlated with PRR, it is a good metric. Some PRR-based link quality estimations are derived from PRR, such as the Window Mean with Exponentially Weighted Moving Average (WM-EWMA), which uses an exponential weighted moving average filter to smooth processing PRR estimation to provide more stable and flexible estimates [10]. At the same time, this method also lays the foundation for other later filter-based link quality estimations. Kalman-filter-based Link quality Estimator (KLE) based on Kalman filter is proposed to overcome the low reactivity of mean-based link quality estimation. Compared with the method waiting to receive a fixed number of packets and then calculate the mean value, the method provides a link quality estimation based on a single received packet. Expected Transmission Count (ETX) is the inverse of the product of forward delivery rate \( d_f \) and reverse delivery rate \( d_r \) [11]:

\[
ETX = \frac{1}{d_f \times d_r}.
\]

ETX-based routing protocols can provide a high throughput route in multihop wireless networks because ETX minimizes the expectation of the total number of packet transmissions required for a packet to be successfully delivered to the destination. The advantage of ETX is that this approach allows for link asymmetry, while the disadvantage is that ETX is ARQ-based, so if the device does not support ARQ, ETX will not be available. Moreover, although ETX can be used to obtain high throughput, when the network traffic load is large, it will lead to network congestion and a large number of packet loss, so a large number of nodes cannot calculate ETX, because they cannot receive packets. Therefore, the lack of link quality information leads to the interruption of routing and the decrease of network throughput.

2.2. Analysis of Existing Wireless Communication Routing Protocol. Flooding protocol and gossiping protocol are classic protocols in sensor networks. They are characterized by the fact that they do not require any algorithms or routing maintenance. The implementation of the flooding protocol is simple, but there are many shortcomings, such as “implosion.” Overlap occurs when two nodes that monitor the same area send similar packets to the same neighbor nodes [12]. The consumption of resources without considering resource constraints leads to the blind utilization of resources. Gossiping protocol improves the flooding protocol, the main difference is that the node randomly selects a neighbor node to send the packet after receiving the packet, and the neighbor node to receive the packet also propagates in this way [13]. This method avoids the problem of “implosion” but leads to the delay of data passing through nodes, and the problem of energy waste caused by random transmission is still unsolved. Sensor Protocols for Information via Negotiation (SPIN) is one of the routing protocols centered on data in early work. The SPIN names the data with high-level descriptors or metadata, but the naming format has no uniform standard, and SPIN also has some shortcomings, such as its announcement mechanism cannot guarantee that the data will be passed. If a node farther from the source node is interested in the data but the nodes between these nodes and the source node are not interested in the data, the data cannot be passed to the destination [14]. As a result, SPIN protocols are not suitable for applications that require high reliable delivery, such as intrusion detection. Rumor protocol is a variant of directed diffusion [15]. In some cases, even if the node needs a very small amount of information, it needs to flood the interest to the whole network. Rumor protocol’s main idea is to route queries to nodes that observe specific events rather than flooding the entire network. However, its good performance is limited to fewer events, and the overhead of maintaining agents and event tables per node becomes large when there are more events. In addition, the cost of adjusting parameters in the algorithm is also a problem. Geographic Adaptive Fidelity (GAF) Protocol is a location-based energy-sensing routing algorithm for mobile self-organizing networks but also for sensor networks [14]. However, the simulation results show that the performance of the GAF is not lower than that of the general ad hoc networks, and it prolongs the network lifetime by energy-saving mechanism. In order to ensure reliable data transmission, Sequential Assignment Routing (SAR) can be used [14]. This is the first routing protocol that introduces QoS into routing decision in wireless sensor network. Its idea is to take the neighbor node of sink node as the root and build a tree considering QoS index, energy on each path, and priority of each packet, so as to build multiple paths from sink node to sensor node [16]. The simulation results show that SAR can achieve lower energy consumption than the minimization energy consumption index algorithm, but it is not suitable for large-scale networks because it needs to maintain multiple paths from nodes to sink directly by maintaining the tables and states of all sensor nodes, so the energy consumption increases greatly when the number of nodes is especially large [14]. Kumar et al. propose a new wireless routing protocol based on two-hop neighbor node information by minimizing path settings, which can improve end-to-end packet delivery recovery delay to ensure reliability and timeliness [17]. SPEED Protocol provides end-to-end soft real-time communication by using an innovative combination of feedback control and uncertain geographic information forwarding to maintain the desired delivery speed, and SPEED Protocol tries to ensure that each packet determines the speed so real-time applications can estimate end-to-end delays before making decisions while avoiding congestion [18]. According to the simulation results in [19], it is clear that SPEED Protocol is deficient in energy efficiency.

3. Reliable Routing Algorithm Based on Link Quality Estimation

3.1. Network Model. The network topology is shown in Figure 1. The role of wireless communication nodes here is
to periodically send their own monitoring data and forward the data sent by other nodes. All nodes send data to the gateway and transmit it to the back-end server through the backbone for analysis and processing. The wireless nodes are battery powered and use the ZigBee protocol. These nodes operate at 2.4 GHz with 16 channels in O-QPSK (Offset-Quadrature Phase Shift Keying) modulation mode. Node antenna adopts ideal omnidirectional antenna, which is easy to analyze and design. The signal has a transmitting power of 0 dBm, a sensitivity of -85 dBm, and a turnaround time of no more than 12 symbol cycles between the transmitting and receiving states. A maximum message length of 127 bytes is ZigBee’s standard upper limit. Because of the periodicity of IWNs, the node monitoring data will also exhibit a corresponding periodicity. We assume that all nodes send packets at the same frequency to facilitate our analysis.

3.2. Selection of Link Quality Estimation. Starting with the link quality analysis based on hardware and software, the link quality estimation introduced earlier is analyzed, and then, the choice of link quality estimation is put forward.

Hardware-based RSSI, SNR, and LQI and software-based PRR and ETX, where RSSI can be read by hardware without additional computational overhead, and the link quality can be immediately judged to be in an excellent or extremely poor state. However, because the value difference between excellent and extremely poor link quality on the RSSI is not obvious, RSSI is not the optimal scheme. SNR is the ratio of signal intensity to noise intensity, so it can reflect the influence of noise more than RSSI. The average LQI can provide a relatively accurate link quality estimate, but an important problem of the LQI is that the stability, variance, or standard deviation are poor when the link quality is medium. Therefore, the value of a single LQI cannot be used to estimate the link quality state. Literature [20] obtained the relationship between average LQI and link delivery rate and the relationship between average LQI and standard deviation of LQI through experimental results. The value of LQI is positively correlated with link reception rate, so this paper believes that LQI should be selected as link quality estimation.

While software-based link quality estimation, PRR and ETX have significantly different characteristics. PRR is more direct to show the probability of successful delivery between links, while ETX needs to be calculated by the delivery rate in two directions, which adds extra overhead. ETX can show the link quality between two nodes by judging the close degree to the upper limit of transmission times. But the traditional ETX ignores the upper limit of retransmission times in the protocol; the effect of Distribution-Based Expected Transmission Count (DBETX) is better than traditional ETX [20]. Considering the ETX of retransmission upper limit, that is, the weighted calculation of average transmission times and SNR distribution, the calculation method is obtained. That is to say, if the traditional ETX calculation results exceed the retransmission upper limit, it is obviously not in line with the actual situation to judge that the link is not reachable. At the same time, the sum of ETX for multi-hop links does not always reflect the link quality or reliability of the whole link. Take Figure 2 as an example.

NodeS is the source node and NodeD is the destination node. Now NodeS can forward to NodeD through node M, or forward to NodeD through node N. P1 is the delivery rate from S to M, P2 is the delivery rate from M to D, P3 is the delivery rate from S to N, and P4 is the delivery rate from N to D. If calculated using PRR, the final success rate of R1 is 1/3 and R2 is 1/4, so choose R1. If ETX or DBETX is used, the sum of ETX in R1 is 4 and the sum of ETX in R2 is also 4, so the performance of the two routes is equal. Obviously, this is not true. The reason is that if PRR is used as link quality estimation, the final success rate can be calculated by multiplying the success rate of each hop on the path, as shown in Formula (2). ETX is the reciprocal sum of the success rate of each jump, as shown in Formula (3).

\[ \prod_{i=1}^{n} p_i > \prod_{j=1}^{m} q_j, \quad (2) \]

\[ \sum_{j=1}^{m} \frac{1}{q_j} < \sum_{i=1}^{n} \frac{1}{p_i}, \quad (3) \]

Formula (2) and Formula (3) in p, and q in the success rate of hop i and hop j on both links; n and m are the total hops on both links. If PRR is used as link quality estimation, then the routing algorithm is based on the return value of Formula (2) to determine which two links are better. If ETX is used as link quality estimation. Then, the return value of Formula (3) is used to estimate. However, Formula (3) cannot be derived from Formula (2). On the contrary, Formula (2) cannot be deduced from Formula (3); hence, ETX cannot well reflect the reliability of the whole path. However, lower expected transmission times indicate that lower total energy consumption can be obtained. So to improve reliability, PRR should be chosen in software-based link quality estimation but PRR and ETX have a limitation that they are both calculated based on the received packets, so they require a certain number of probe packets. Sometimes, the field application of IWNs does not allow a large amount of time to send a sufficient number of probe packets.
 packets before the formal operation, and the performance based on the received link quality estimation will decrease. Therefore, this paper thinks that the link quality estimation based on hardware and the link quality estimation based on software should be combined. Although in literature [21], the related work is summarized to show that there is no significant correlation between RSSI, LQI, and ETX, but the literature [9] points out that the average LQI has some correlation with PRR. Particularly, the literature [22] proposed a curve fitting formula for average LQI and PRR, as shown in Formula (4). Through this formula, the value of a PRR can be estimated by the LQI value and used in routing calculation and the calculation basis of packet loss in simulation. And the LQI calculation method can be based on the literature [23]. A linear regression is provided to derive Formula (5). Even if the hardware itself does not support computing LQI, it can be obtained by SNR. Furthermore, although the negative effects of LQI instability can be reduced by averaging, recording a value from a probe packet from a neighbor node for averaging consumes a lot of storage space. Therefore, it is necessary to take some methods to reduce the consumption of storage space and energy.

![Image](image.png)

**Figure 2:** PRR Comparison with ETX on routing.

\[
P_{\text{RRR}} = \begin{cases} 
95\%_{\text{RSSI}} < 65 - 0.0001577694347132523LQI^3 + 4.708425273599874LQI - 214.9115458068067 \times (\%) & 65 \leq \text{RSSI} \leq 95, \\
5\%_{\text{RSSI}} > 95, 
\end{cases}
\]

\[
LQI = 5.3145 \times \text{SNR} + 94.0477. 
\]

An optimal linear filter, Kalman filter, is used here to filter and smooth, which can make an accurate estimate from the nonmeasurable state of a dynamic system with observed noise [24]. Therefore, in the instability modeling of LQI in this paper, white noise is added according to its real value. The advantage of this approach is that the error can be minimized through Kalman filtering, thus improving the accuracy. Because the input data has only LQI values, a one-dimensional Kalman filter is sufficient here, and its implementation is presented in Algorithm 1. The algorithm can be simply described as follows: input data, use the least square method to estimate the minimum variance, and then output the results, iterating repeatedly.

**3.3. Topology Discovery.** Before the routing algorithm runs, some necessary information should be collected and used in routing calculation. Topology discovery initiated by the gateway. The depth of the gateway node is 0, and the depth of the other nodes is set to the desirable maximum. Then, the gateway generates a probe packet to get depth and broadcasts it. These nodes received are neighbor nodes and compare the depth value in the probe packet with their existing depth, if the former is smaller, set its own depth information to the former, then generate a new probe packet, and set the depth to the updated depth before broadcasting; otherwise, no behavior is taken. In this way, when the probe packet flooding to the whole network, all nodes can get their own depth relative to the gateway. At the same time, we also know the neighbor nodes and their depth information. The probe packet format used to obtain depth information is shown in Table 1.

Among them, type is the type of message, which is used to distinguish different kinds of messages for different processing. Field source is the address of the source node that sends the message, or the node number can be used to know the source of the message. Field depth is the depth of the node that sends the message, the node that receives the message determines whether it can obtain a smaller depth through this field, where the depth of the gateway is 0. By receiving and sending messages, the node cannot only obtain its own depth but also make all nodes in the network obtain the corresponding information. At the same time, when the node obtains its depth relative to the gateway, the next hop of the minimum hop routing can be obtained at the same time. However, because this process needs to flood the depth probe packet from the gateway node to the whole network, it needs to consume more energy. However, the minimum hop routing is also obtained. Although the minimum hop routing does not guarantee any QoS and reliability, there is also a default route available without other routes. In addition, the depth of a node may change many times during the probe of packet flooding, and the depth of a node may change the depth of some neighbor nodes. Therefore, it is necessary to update the depth probe packet flooding in real time.

**3.4. Access to Link Quality-Related Information.** After obtaining the depth of the node itself relative to the gateway, the link quality is estimated by sending the probe packet. In
3.5. Algorithm

3.5.1. Algorithm. Through the above analysis, we finally determine the specific method of reliable routing-based link quality estimation (HLQEBRR). First, topology discovery and link quality information collection are carried out by sending probe packets, and then, the corresponding information is recorded and processed. When the algorithm runs, the gateway node starts to send the routing packet to the neighbor node, and the neighbor node receives the routing packet and runs the algorithm to find out the success rate, the next hop, the path length, and the threshold of failure monitoring. The routing packet is sent to the neighbor node the routing packet is sent to the neighbor node, and the neighbor node receives the routing information package is shown in Table 4.

The routing packet generation algorithm is shown in Algorithm 3. The source next hop in the routing packet first constructed by the gateway node are all their own addresses, the success ratio is 1, and the path length is 0. After the node receives, the routing algorithm is run, and the result of the routing calculation is informed to the neighbor node.

On the basis of Formula (4), an approximate LQI can be obtained by curve fitting PRR, which is called PRR_{LQI}. A PRR can also be obtained based on the ratio of the number of received packets to the total number of probe packets sent. In this case, the weighted average of the two PRRs should be carried out to obtain the mean value of a PRR, and the weight should be determined according to the total number of probe packets. Since it is an approximation obtained by curve fitting, the weight of PRR_{LQI} obtained by curve fitting should decrease with the increase of the number of probe packets. Therefore, Algorithm 4 can be obtained to calculate a weighted average PRR value. The algorithm of calculating packet receiving rate according to link quality information is shown in Algorithm 4.

**Algorithm 1: One-dimensional Kalman filter algorithm.**

```plaintext
Require: the value of the LQI with additive Gaussian white noise and the result of the last filter and other parameters of the Kalman filter
Ensure: filtered results
1: initialize: \( p \leftarrow 1; R \leftarrow 1; LQI \leftarrow \) the result after the last filtering
2: while value of LQI with noise \( X \) read in do
3: \( K \leftarrow (P + R) \)
4: \( LQI \leftarrow LQI + K \times (X - LQI) \)
5: \( P \leftarrow P - K \times (P + R) \times K \)
6: end while
7: return \( LQI \)
```

**Table 1: The probe packet format used to obtain depth.**

| Field   | Description                        |
|---------|------------------------------------|
| Type    | Message type                       |
| Source  | Address of the source node that sent the message |
| Depth   | The depth of the source node that sends the message |
| LQI     | Link quality indicator             |
| \( P \) | A posteriori estimate covariance in Kalman filtering |

**Table 2: Format of link quality probe packet.**

| Field   | Description                        |
|---------|------------------------------------|
| Type    | Message type                       |
| Source  | Address of the node that sent the packet |
| Total   | Total number of packets sent       |
| Seq     | Serial number of the current probe packet |

by reconstructing its own results with the collected link quality information, until all nodes in the network are routed. The record format of link quality information collection is shown in Table 3. The algorithm of link quality information collection, such as Algorithm 2, is also processed while collecting data, where the Kalman filter runs with the process of link quality collection. Each new value is obtained by running a Kalman filter to obtain the filtered data, except for the first obtained value.

When the node sends the link quality probe packet, the routing algorithm can be initiated by the gateway. At this point, the gateway constructs the routing information packet shown in Table 4, constructs the routing information packet, and then broadcasts it to the neighbor node. The format of the routing information package is shown in Table 4.

The routing packet generation algorithm is shown in Algorithm 3. The source next hop in the routing packet first constructed by the gateway node are all their own addresses, the success ratio is 1, and the path length is 0. After the node receives, the routing algorithm is run, and the result of the routing calculation is informed to the neighbor node.
The main function of the routing algorithm is to make all nodes find the optimal path to send messages, so all nodes need to obtain the end-to-end transfer rate obtained through all possible paths. The link quality estimation used in the routing algorithm designed in this paper is that both PRR and LQI are the data obtained by the receiving end and the sending end does not know it. Therefore, the routing calculation can be carried out by broadcasting to the neighbor node to obtain its own delivery rate corresponding to the node. In addition, in the routing mechanism of this paper, the hopping confirmation retransmission mechanism is used to improve the reliability. Suppose the maximum number of transmissions in a network with a confirmation retransmission mechanism is $n$, nodeA already knows the PRR of nodeB for nodeA is $p$; then, the probability $P_{\text{delivery}}$ of nodeA successfully sending packets to nodeB is

$$ P_{\text{delivery}} = 1 - (1 - p)^n. \quad (6) $$

The partial derivative of $P_{\text{delivery}}$ with respect to $p$ and $n$ can be obtained:

$$ \frac{\partial P_{\text{delivery}}}{\partial p} = n(1 - p)^{n-1}. \quad (7) $$

### Table 3: Format for storing link quality information.

| Field | Description |
|-------|-------------|
| Address | Address of neighbor node |
| Sum | Total number of probe packets sent by neighbor nodes |
| Count | Number of probe packets received from the neighbor node |
| LQI | Link quality indicator |
| $P$ | A posteriori estimate covariance in Kalman filtering |
| $R$ | Covariance of observed noise in Kalman filtering |
| $K$ | Kalman gain in Kalman filtering |

### Algorithm 2: Link quality information collection algorithm.

```plaintext
Require: Numeric X of probe packets and read LQI from neighbor nodes
Ensure: Information for link quality estimation of neighbor nodes
1: \textbf{while} address ≠ source \textbf{do}
2: \textbf{if} has next record \textbf{then}
3: \hspace{1em} sum ← total
4: \hspace{1em} count ← count + 1
5: \hspace{1em} KalmanFilter(LQI, X, P, R, K)
6: \textbf{else}
7: \hspace{1em} sum ← total
8: \hspace{1em} LQI ← X
9: \hspace{1em} P ← 1
10: \hspace{1em} R ← 1
11: \hspace{1em} K ← 0.5
12: \textbf{end if}
13: \textbf{end while}
```

### Table 4: Format of routing information packet.

| Field | Description |
|-------|-------------|
| Source | Source address |
| Nexthop | The address of the next hop of the node route |
| Success_ratio | The success rate of the node passing the packet to the gateway |
| Path_length | Path length |
| Neighbor_num | Number of neighbor nodes |
| Address_info[] | An array of neighbor node addresses |
| PRR[] | Arrays of packet reception rates for different neighbor nodes |
| Repair | Mark whether the routing recovery process is needed |
At the same time, the error caused by the link quality estimation and the increase of the maximum transmission times. Therefore, the impact of the retransmission mechanism, where the higher the success rate, the smaller the impact of the retransmission mechanism, At the same time, it shows that the lower the success rate, the greater the impact of the retransmission mechanism. Formula (7) shows that if \( \alpha < 1 \), the results of both partial derivatives are nonnegative. Formula (8) shows that when the \( p \) is constant, the value of the partial derivative decreases with the increase of the \( n \), which indicates that the increase in delivery rate decreases with the increase of the maximum transmission times. At the same time, the effect of error decreases with the increase of maximum transmission times. Therefore, the hop-by-hop retransmission mechanism cannot only directly improve the reliability of transmission but also reduce the impact caused by the error of link quality estimation and further improve the reliability. The design goal of the HLQEBRR algorithm is to maximize the reliability of all nodes. Based on literature [25], the concept of the most reliable line is that if \( \forall e \in E(G) \) in graph \( G(0 < w(e) \leq 1) \). If \( P \) is the path from vertex \( S \) to vertex \( T \) and is defined as the reliability of path \( P \), then the path that maximizes \( w(P) \) is the optimal path from vertex \( S \) to vertex \( T \). The optimal path is the path \( P \) from vertex \( v_{i,j} \) to a certain victorious vertex \( v_{j,k} \) in the decision graph \( D = (V, E) \). If the cost is the least among all victorious vertices from \( v_{i,j} \) to \( v_{j,k} \), then \( P \) is the optimal path of vertex \( v_{i,j} \). According to the optimization principle, let \( P = v_{i,j}v_{i+1,j}v_{i+2,j} \cdots v_{j-1,j}v_{j,k} \) is an optimal path for vertex \( v_{i,j} \), then \( P = v_{i,j}v_{i+1,k}v_{i+2,k} \cdots v_{j-1,k}v_{j,k} \) must be the optimal path for vertex \( v_{i,j} \). Therefore, the HLQEBRR algorithm proposed in this paper is initiated by the gateway and sends the routing information packet that carries the PRR value and its source information obtained by the probe packet sent by the neighbor node, plus their own calculated routing results. After receiving the neighbor node, the product of the success rate of the source node and the success rate of sending packets to the node is calculated and recorded. Repeat the above process if the value is better than the

**Algorithm 3: Generation algorithm of routing information packet.**

```
Require: Collection of link quality information
Ensure: Routing packets
1: source <-- address
2: nexthop <-- optimal_nexthop
3: success_rate <-- optimal_success
4: path_length <-- optimal_hop+1
5: neighbor_num <-- n_num
6: repair <-- false
7: i <-- 0
8: while do
9:   if alive = true then
10:      address_info <-- address[i]
11:      PRR[i] <-- calculate(i)
12:   end if
13:   next p
14: end while
```

**Algorithm 4: Package reception rate estimation.**

\[
\frac{\partial P_{\text{delivery}}}{\partial n} = -(1-p)^n \ln (1-p) = (1-p)^n \ln \frac{1}{1-p}.
\]

(8)
previous solution, update the optimal solution, and broadcast the routing packet in the same way. Until all nodes in the network receive routing packets from all their neighbor nodes and complete the calculation of the optimal path, the algorithm is aborted. When the routing algorithm updates the optimal solution, the condition of the update is not just that the end-to-end delivery rate of the new path is higher than the previous solution, instead, consider weighing the negative cost before updating. Although the end-to-end delivery rate is slightly improved, some solutions increase delay and energy consumption at great cost. Each node initializes the probe packet first and sets the table of neighboring nodes as an empty set. After receiving the probe packet, if the depth value increased by 1 is less than its own depth value, the current depth value will be updated. The probe packet is reconstructed and forwarded to the neighbor node. After receiving the probe packet for acquiring depth, the node can send the link quality information probe packet, which is used to make the neighbor nodes obtain the packet reception rate relative to its own, and the neighbor nodes run Algorithm 3 to obtain and store the corresponding information.

Writes its own node address to source, calculate the next hop, assign the success rate to nexthop, success_ratio separately, number of neighbor nodes written neighbor_num, an array address_info[] that traverses the link quality information list to supplement the addresses of neighboring nodes, calculate packet acceptance rate based on link quality information, and write the array PRR[] with Packet Reception Ratio for different neighboring nodes. In the routing packets first constructed by the gateway node, source and nexthop are their own addresses, success_ratio is 1, and path_length is 0. After receiving it, the node runs the routing algorithm and informs neighbor nodes of the routing calculation. The HLQEBRR algorithm is shown in Algorithm 5.

3.5.3. Node Failure Treatment. The probability-based method is used to detect node failure, that is, small probability events cannot occur in an experiment. Set the nodeA has learned the delivery rate between it and the nodeB and set it to \( p \), the maximum number of transmissions. At the same time, the node sets a counter and checks the number of packets sent at this time when receiving the confirmation packet. If the maximum number of transmissions is reached, the counter adds 1; then, the probability \( P_{\text{fail}}(n) \) of sending \( n \) consecutive data packets with the maximum transmission times but not receiving the acknowledgement packet from nodeB is

\[
P_{\text{fail}}(n) = ((1-p)^\text{max})^n.
\]

If \( P_{\text{fail}}(n) \) is less than a threshold, such as 0.00001, then nodeB is deemed to have failed. In this case, \( N \) is the threshold value of the sent packet that does not receive the acknowledgement packet in the node failure monitoring process and adopts the node failure recovery mechanism. For the node failure recovery mechanism, this paper adopts such a mechanism to record the suboptimal solution while calculating the routing optimal solution. When the optimal solution is updated, the previous optimal solution is regarded as the suboptimal solution until the path calculation is completed. The suboptimal solution is the standby path. When the next hop node fails, it is first switched to the standby path, then the node failure occurs along the downlink notification node and the node on the link is recalculated. For the nodes of the failed nodes along the uplink, the existing calculation results are still available and there is no need to reroute because their path of sending packets is not affected. If the node is restored after node failure, the routing control package can be sent again by the neighbor node according to the previous routing algorithm. The node failure recovery mechanism is shown in Algorithm 6.

4. Simulation and Results

The hardware environment and simulation environment are shown in Table 5.

In this section, the simulation implementation of the HLQEBRR algorithm is introduced in detail, and the performance of sending different numbers of probe packets is evaluated. The comparison benchmark algorithm used in performance evaluation is the Collection Tree Protocol (CTP), and the results are analyzed.

4.1. Simulation Environment and Parameter Setting. There are three types of nodes in the network topology set up in this paper, which are industrial field communication equipment nodes, gateway nodes, and noise nodes. The simulation topology used in this paper is shown in Figure 3. The industrial site is a rectangle, and the field equipment nodes are randomly distributed everywhere. The two-way arrow indicates the connection and has been selected according to the maximum communication range so that any node connected with a two-wire arrow is within the range of communication. While the communication range of noise nodes is not limited, all field device nodes and gateways are connected with them. The specific parameters of the simulation topology are shown in Table 6.

4.2. Evaluation Indicators and Comparison Benchmark

4.2.1. Evaluation Indicators

(1) End-to-end delivery and packet loss rates

Assume that node, in the network sends total, packets, and the number of packets successfully transmitted to the destination is delivery; then, the lost number is total, minus delivery. At this point, the end-to-end delivery rate \( r_{\text{delivery}}(i) \) and packet loss rate \( r_{\text{loss}}(i) \) of node \( i \) are defined

\[
r_{\text{delivery}}(i) = \frac{\text{delivery}_i}{\text{total}_i},
\]

\[
r_{\text{loss}}(i) = 1 - \frac{\text{delivery}_i}{\text{total}_i}.
\]

The average end-to-end network delivery rate \( r_{\text{delivery}} \)
Require: Routing information packets from neighbor nodes
Ensure: Success rate of sending packets to the gateway, address and path length of the next hop node
1: \( \text{optimal\_success} \leftarrow 0 \)
2: \( \text{optimal\_nexthop} \leftarrow \text{nexthop} \)
3: \( \text{optimal\_hop} \leftarrow \text{mydepth} \)
4: if \( \text{repair} = \text{true} \) and \( \text{source} \neq \text{optimal\_nexthop} \) or \( \text{source} \neq \text{suboptimal\_nexthop} \) then
5: return
6: end if
7: for \( i = 0, 1, 2, \ldots, \text{neighbor\_num} \) do
8: if \( \text{address\_info}[i] = \text{address} \) then
9: \( \text{suc} \leftarrow 1 - (1 - \text{PRR}[i])^{\text{maxtrans}} \times \text{success\_ratio} \)
10: According to the \text{source} field of the routing packet, find the corresponding record in the routing result and update it. If not found, record the \text{source}, \text{suc}, \text{nexthop}, \text{pathlength} value of the routing packet
11: if \( \text{optimal} \neq \text{source} \) then
12: \( \text{suboptimal\_nexthop} \leftarrow \text{optimal\_nexthop} \)
13: \( \text{suboptimal\_success} \leftarrow \text{optimal\_success} \)
14: \( \text{suboptimal\_hop} \leftarrow \text{optimal\_hop} \)
15: \( \text{suboptimal\_threshold} \leftarrow \text{optimal\_threshold} \)
16: end if
17: \( \text{optimal\_nexthop} \leftarrow \text{source} \)
18: \( \text{optimal\_success} \leftarrow \text{suc} \)
19: \( \text{optimal\_hop} \leftarrow \text{pathlength} + 1 \)
20: \( \text{P}_{\text{fail}} \leftarrow (1 - \text{PRR}[i])^{\text{maxtrans}} \times \text{success\_ratio} \)
21: \( \text{optimal\_threshold} \leftarrow 1 \)
22: while \( \text{P}_{\text{fail}} > 0.00001 \) do
23: \( \text{P}_{\text{fail}} \leftarrow (1 - \text{PRR}[i])^{\text{maxtrans}} \times \text{success\_ratio} \)
24: \( \text{optimal\_threshold} \leftarrow \text{optimal\_threshold} + 1 \)
25: end while
26: generate routing packets according to Algorithm 3 and inform neighbor nodes
27: end if
28: end for

Algorithm 5: HLQEBRR algorithm based on link quality estimation.

Require: Failure confirmation information during node failure detection
Ensure: Information on alternate paths and new routing information
1: while do
2: if \( \text{p} \rightarrow \text{address} = \text{optimal\_nexthop} \) then
3: \( \text{p} \rightarrow \text{alive} \leftarrow \text{false} \)
4: \( \text{n\_number} \leftarrow \text{n\_number} - 1 \)
5: end if
6: \( \text{p} \leftarrow \text{p} \rightarrow \text{next} \)
7: end while
8: if the suboptimal solution exists and is not the solution of the minimum hop route then
9: \( \text{optimal\_nexthop} \leftarrow \text{suboptimal\_nexthop} \)
10: \( \text{optimal\_success} \leftarrow \text{suboptimal\_success} \)
11: \( \text{optimal\_hop} \leftarrow \text{suboptimal\_hop} \)
12: \( \text{optimal\_threshold} \leftarrow \text{suboptimal\_threshold} \)
13: else
14: reverse the routing result record to find the solution with the highest success rate and regarded it as the current optimal solution
15: end if
16: generate routing packets according to Algorithm 3
17: \( \text{repair} \leftarrow \text{true} \)
18: sent routing packets to neighboring nodes
19: return \( \text{optimal\_nexthop} \)

Algorithm 6: Node failure recovery mechanism.
and packet loss rate $r_{\text{loss}}$ is

$$r_{\text{delivery}} = \frac{\sum_{i=1}^{n} \text{delivery}_i}{\sum_{i=1}^{n} \text{total}_i},$$

$$r_{\text{loss}} = 1 - \frac{\sum_{i=1}^{n} \text{delivery}_i}{\sum_{i=1}^{n} \text{total}_i}.$$  \hspace{1cm} (12) \hspace{1cm} (13)

(2) Network average end-to-end delay

The transmission time from node $i$ to node $j$ is $t_{Sj}(j)$, and the time for node $i$ to reach the gateway to be received is $t_{Si}(j)$; then, the average end-to-end delay $\text{delay}(i)$ of node $i$ and the average network end-to-end delay are, respectively:

$$\text{delay}(i) = \frac{\sum_{j=1}^{n} t_{Di}(j) - t_{Si}(j)}{\text{total}_i},$$

$$\text{delay}(i) = \frac{\sum_{i=1}^{n} R(i)}{n}. \hspace{1cm} (14) \hspace{1cm} (15)$$

(3) Overall network energy consumption

Only the energy consumption generated by sending packets after the routing algorithm is completed is considered here, because this part is the most important part of the overall energy consumption of the network, while the energy consumption generated by sending probe packets, routing packets, and computing processes in the routing algorithm is ignored. Because the direct calculation of power consumption is not feasible and the simulation is used here, the overall energy consumption of the network is reflected by the number of packets sent, so Formula (16) is used to reflect the overall energy consumption of the network.

$$\text{trans\_num} = \sum_{i=1}^{n} \text{trans\_num}(i). \hspace{1cm} (16)$$

The $\text{trans\_num}(i)$ is the number of packets sent by node $i$. In order to simplify the analysis and facilitate the performance comparison, the number of received packets and sent and received confirmation packets is no longer calculated here, and the energy of the node is assumed to be infinite.

(4) Reliability

The measure of reliability here is the ratio of nodes with end-to-end delivery rate above 95% or packet loss rate below 5% to the total number of nodes in the network. The expression is as follows:

$$\text{reliability} = \frac{\sum_{i=1}^{n} \text{R}(i)}{n},$$

$$R(i) = \begin{cases} 1 & r_{\text{delivery}}(i) \geq 95\%, \\ 0 & r_{\text{delivery}}(i) < 5\%. \end{cases} \hspace{1cm} (17)$$

4.2.2. Comparison Benchmark. The comparison benchmark adopted in this paper is the Collection Tree Protocol (CTP) [26], which is designed to meet the requirements of reliability, robustness, energy efficiency, and hardware independence; it is the sink route to one or a few specified nodes in a wireless sensor network. The CTP used link quality estimation of 4 bits; routing is based on the sum of the expected transmission times on the whole path and only 1% lower than the previous minimum ETX of the sum of the new path. The new path is used as a new route. CTP is divided into three parts, link quality estimation, routing engine, and forwarding engine. The link quality estimation is used to estimate the number of single hops expected to be transmitted to the neighbor nodes. The routing engine is the next hop of routing according to the link quality and network layer information, and the forwarding engine [26]. It is used to maintain the queue of packets waiting to be sent and whether or not to send them, so the CTP algorithm is very suitable as a comparison benchmark. Because the HLQEBrR algorithm and the CTP algorithm proposed in this paper need to send probe packets first, the performance of each algorithm can be analyzed when different numbers of probe packets are sent in advance. The simulation parameters are shown in Table 7.
4.3. Analysis of Experimental Results. First 30 probe packets are sent, and the performance results of the two algorithms are shown in Table 8. MATLAB programming was used to read the files of these records and calculate the related performance indicators. After the calculation is completed, the drawing is carried out according to the result of the operation.

The performance of each node can reflect the performance of the routing algorithm. Such as the packet loss rate for each node shown in Figure 4, the end-to-end delay for each node is shown in Figure 5, and the number of sendings for each node is shown in Figure 6.

The HLQEBRR algorithm proposed in this paper has some advantages in the average delivery rate of the network, but it is slightly insufficient in the average delay and the overall transmission times. However, from the performance of each node, the routing algorithm proposed in this paper can maximize the delivery rate of each node. According to Formula (15), the reliability of this paper is 0.95 higher than 0.75 of the CTP algorithm. At the same time, other performance deficiencies are not significant.

The end-to-end delay of each node can be found that even if the CTP algorithm is based on the sum of the expected transmission times on the path, end-to-end delays include other delays besides sending delays, such as queuing delays. Therefore, the CTP algorithm has some advantages over the HLQEBRR algorithm in average end-to-end delay, but it is not obvious, even cannot guarantee that the end-to-end delay performance of each node is better than the HLQEBRR algorithm. At the same time, it also has some advantages in the index of sending times, because the route selected by smaller expected transmission times reduces the energy cost of sending messages. Run the simulation again for performance comparison, and the number of probe packets is increased to 100. The overall performance of the network is shown in Table 9. The packet loss rate, latency, and number of times of each node are shown in Figures 7–9.

It can be seen that both the HLQEBRR algorithm and the CTP algorithm have achieved better performance after sending more probe packets. That is to say, the number of probe packets will have a certain impact on the algorithm of selecting the next hop routing through link quality estimation. We can see that when the number of probe packets increases, the HLQEBRR algorithm and the CTP algorithm proposed in this paper improve the performance index of the delivery rate. This is because when the number of probe packets increases, both the ratio of the received probe packet to the total amount and the packet reception rate obtained by curve fitting are more accurate than when there are fewer

Table 7: Simulation parameters.

| Parameter name                               | Value                        |
|----------------------------------------------|------------------------------|
| Simulation time                              | Until all events are completed|
| Delay of node transmitting packet            | 0.01 s                       |
| Delay of node sending confirmation packet    | 0.0025 s                     |
| Number of sounding packets sent              | 30/100                       |
| Number of packets sent                       | 100                          |
| Maximum number of repetitions                | 4                            |
| Noise power of noise nodes                   | 10 dBm                       |

Table 8: Performance of entire network.

| Routing mechanisms | Average delivery rate | Average end-to-end delay | Number of total packets |
|--------------------|-----------------------|--------------------------|-------------------------|
| HLQEBRR            | 99.5%                 | 0.0965 s                 | 15193                   |
| CTP                | 90.7%                 | 0.0874 s                 | 14690                   |

Figure 4: Packet loss ratio of every node.

Figure 5: End-to-end delay of every node.
probe packets. Therefore, the performance of link quality estimation will be significantly improved.

The HLQEBRR algorithm proposed in this paper uses packet receiving rate as link quality estimation CTP compared with the expected transmission times as link quality estimation and routing standard. Because the product of link quality and hop-by-hop PRR is more accurate and intuitive, the idea of optimal path HLQEBRR adopted in this paper ensures that every node in the network can obtain the optimal path through the result of link quality estimation (not pursuing the highest delivery rate, avoiding too long path, and improving performance). Therefore, the HLQEBRR algorithm proposed in this paper is obviously superior to the CTP algorithm in the delivery rate of each node, so the overall delivery rate of the HLQEBRR algorithm will be higher than that of the CTP algorithm from the point of view of the whole network. The reliability of the proposed HLQEBRR algorithm is 1.00, and that of the CTP algorithm is 0.825. The average end-to-end delay is very similar to the average end-to-end delay index of the network, which is caused by the delay that needs to be retransmitted due to packet loss and the queue delay that waits for the packet to be sent even after it is received. The CTP algorithm has no obvious advantage in delay because the sum of ETX on the path only reflects the delay caused by retransmission due to packet loss, but it does not reflect queue delay. At the same time, the HLQEBRR proposed in this paper has some disadvantages in terms of delay, but it is also due to the successful delivery of more packets to the gateway, which increases the queue delay of other nodes on each hop link. Therefore, the proposed HLQEBRR algorithm has some disadvantages in delay, but it is not obvious. Since the CTP algorithm is to minimize the sum of the transmission times of the whole path, the CTP has a very significant advantage in energy efficiency. In contrast, the HLQEBRR proposed in this paper is still inferior in energy consumption. But partly because more packets are successfully delivered to the node, resulting in more transmission times to bring energy consumption. The HLQEBRR proposed in this paper CTP compared with the additional extremely small 1.2% additional energy consumption in exchange for 5.3% significant delivery rate performance improvement and greatly improve

### Table 9: Performance of entire network with 100 probe packets.

| Routing mechanisms | Average delivery rate | Average end-to-end delay | Number of total packets |
|--------------------|-----------------------|--------------------------|-------------------------|
| HLQEBRR            | 99.9%                 | 0.0858 s                 | 14253                   |
| CTP                | 94.9%                 | 0.0800 s                 | 14086                   |

![Figure 6: Transmission times of every node.](image)

![Figure 7: Packet loss ratio of every node.](image)

![Figure 8: End-to-end delay of every nodes.](image)
reliability, it can be considered that the additional energy cost is valuable. We can see from the above data that the HLQEBRR proposed in this paper has significant advantages and disadvantages in reliability compared with CTP, and the performance reduction is relatively small when the number of probe packets is small. Therefore, it can be considered that the HLQEBRR routing proposed in this paper is better than CTP. In selecting transmission paths in addition to the reliable routing, the performance of node failure recovery is compared in the following. In order to compare the performance of the node failure recovery mechanism itself more accurately, three nodes are selected to fail at the beginning of the simulation to simulate the performance of the two routing mechanisms when these three nodes do not exist in the network. Then, the three nodes send the probe packet and then fail before sending the packet. The purpose is to let the other nodes select the three nodes as the next jump to test the performance of the node failure recovery mechanism. At the same time, the delivery rate in this case is the maximum of the delivery rate that the routing algorithm can obtain after failure. In this part, the main focus is on the change of delivery rate. First, node 3, node 12, and node 36 are selected to make them fail at the beginning, so that other nodes do not choose the three nodes as the next hop to avoid the failed link. This will not lead to the selection of these three nodes as the next hop node when sending packets because of the sudden failure of the previously calculated routing, which can be used to obtain the upper limit of the theoretical performance of the node failure processing mechanism. As shown in Figure 10, a node with a packet loss rate of 100% is the selected node, and the overall performance of the HLQEBRR and CTP network is shown in Table 10.

Then, let the selected node complete the task of sending the probe packet and then fail when sending the packet. The two routing mechanisms do not run the node failure recovery mechanism to test the most serious consequences of node failure. At the same time, this result is the lower performance limit of the node failure recovery mechanism. The result of node packet loss rate is shown in Figure 11. The overall network performance of the two routes is shown in Table 11.

It can be seen that the delivery rate of the two routing mechanisms is significantly reduced without node failure recovery. Although the average end-to-end delay and energy consumption are also significantly reduced at this time, the reason is that the packets successfully delivered to the gateway are significantly reduced. Therefore, the end-to-end delay and the improvement of the overall energy consumption index of the network do not indicate the improvement
of the performance of the routing algorithm at this time. It also proves that the weak disadvantage of the routing algorithm proposed in this paper is not due to performance. Then, test the performance of the node in midway failure but perform node failure recovery. The packet loss rate for each node that fails but fails to recover is shown in Figure 12.

Comparing Figures 10–12 and Tables 10 and 11, it can be found that node failure recovery can lead to a large number of nodes losing packets in the downlink direction of the same path in the network. After the node failure recovery mechanism is implemented, the packet loss rate except the failure node is reduced. Therefore, both routing mechanisms effectively improve the reliability. The HLQEBRR algorithm proposed in this paper is better than the CTP algorithm, although the performance of the average end-to-end delay is less than CTP; the disadvantage is negligible. At the same time, it can be determined that although energy consumption is a significant disadvantage, but much of this is due to the energy consumption generated by more successfully delivered packets. Therefore, the simulation results show that the HLQEBRR algorithm proposed in this paper is superior to the CTP algorithm performance of node failure recovery.

5. Conclusion

In order to improve the reliability of data communication in IWNs, this paper proposes a hybrid reliable routing algorithm based on link quality estimation and proposes a link quality estimation based on hardware link quality estimation and software link quality estimation. For the hardware-based link quality estimation LQI, Kalman filter is used to reduce the variance of the estimation, which also reduces the storage space. For software-based link quality estimation, PRR is adopted as software-based link quality estimation after analyzing and comparing the performance of PRR and ETX. This estimation method can not only reduce the calculation amount but also express the link quality estimation more directly, so it is more suitable for routing calculation. Though the analysis of graph theory, a reliable routing algorithm based on link quality estimation based on the idea of optimal path is proposed. Meanwhile, this paper considers the possibility of node failure and judges node failure according to probability, that is, the node failure is judged by calculating whether the probability that the node has not received the confirmation packet for several consecutive times reaches the threshold value. After a node fails, the backup path in the routing algorithm is adopted immediately. At the same time, other nodes are informed along the downlink direction of the path that the failure has occurred, and local rerouting is performed. However, the uplink path does not need rerouting, thus reducing the influence on the network and unnecessary energy consumption. The experimental results show that the routing algorithm with reliability basis combined with link quality estimation can significantly improve the routing performance and data transmission reliability and can respond quickly after the occurrence of failure and reduce the impact of failure on the network.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Projects (2019YFB1802600), the Liaoning Province Science and Technology Fund Project (2020MS0086), the Shenyang Science and Technology Plan Project (20206424), the Fundamental Research Funds for the Central Universities (N2116014 and N180101028), the National Natural Science Foundation of China (62072094 and 61872073), and the CERNET Innovation Project (NGII20190504).

References

[1] G. P. Hancke, V. C. Gungor, and G. P. Hancke, ”Guest editorial special section on industrial wireless sensor networks,” IEEE Transactions on Industrial Informatics, vol. 10, no. 1, pp. 762–765, 2014.
[2] Y. Wu, W. Zhang, H. He, and Y. Liu, ”A new method of priority assignment for real-time flows in the wireless short network by the TDMA protocol,” Sensors, vol. 18, no. 12, p. 4242, 2018.
[3] J. Zhu, Y. Zou, and B. Zheng, ”Physical-layer security and reliability challenges for industrial wireless sensor networks,” IEEE Access, vol. 5, pp. 5313–5320, 2017.
[4] M. C. Lucas-Estan, B. Coll-Perales, and J. Gozalvez, ”Redundancy and diversity in wireless networks to support mobile industrial applications in industry 4.0,” IEEE Transactions on Industrial Informatics, vol. 17, no. 1, pp. 311–320, 2021.
[5] J. Hong, J. Heo, and Y. Cho, ”EARQ: energy aware routing for real-time and reliable communication in wireless industrial sensor networks,” IEEE Transactions on Industrial Informatics, vol. 5, no. 1, pp. 3–11, 2009.
[6] J. Zong, S. Li, D. Zhang et al., ”Smart user pairing for massive MIMO enabled industrial IoT communications,” in 39th IEEE Conference on Computer Communications, INFOCOM Workshops 2020, pp. 207–212, Toronto, ON, Canada, 2020.
[7] N. H. Nguyen and M. K. Kim, ”Link quality estimation from burstiness distribution metric in industrial wireless sensor networks,” Energies, vol. 13, no. 23, article 6430, 2020.
[8] P. Tusset-Beiro, R. D. Gomes, P. Thubert, and X. Vilajosana, ”Evaluating IEEE 802.15.4g SUN for dependable low-power wireless communications in industrial scenarios,” Preprint, 2020.
[9] C. A. Boano, T. Voigt, A. Dunkels et al., ”Poster abstract: exploiting the LQI variance for rapid channel quality assessment,” in Proceedings of the 8th International Conference on Information Processing in Sensor Networks, IPSN 2009, pp. 369–370, San Francisco, California, USA, 2009.
[10] S. Rekik, N. Baccour, M. Jmaiel, and K. Drira, ”Wireless sensor network based smart grid communications: challenges, protocol optimizations, and validation platforms,” Wireless Personal Communications, vol. 95, no. 4, pp. 4025–4047, 2017.
[11] E. D. Spyrou and D. K. Mitrakos, “Etx-based relay selection coalition game for wireless sensor networks,” in 13th International Wireless Communications and Mobile Computing Conference, IWCMC 2017, pp. 705–710, Valencia, Spain, June 2017.

[12] C. Salim and N. Mitton, “Image similarity based data reduction technique in wireless video sensor networks for smart agriculture,” in AINA-2021 35th International Conference on Advanced Information Networking and Applications, Toronto, Canada, 2021.

[13] G. Xiao, N. Sun, L. Lv, J. Ma, and Y. Chen, “An heed-based study of cellclustered algorithm in wireless sensor network for energy efficiency,” Wireless Personal Communications, vol. 81, no. 1, pp. 373–386, 2015.

[14] A. A. Kumar, K. Ovsthus, and L. M. Kristensen, “An industrial perspective on wireless sensor networks - a survey of requirements, protocols, and challenges,” IEEE Communications Surveys & Tutorials, vol. 16, no. 3, pp. 1391–1412, 2014.

[15] C. Raman, L. J. Ali, N. Gobalakrishnan, and K. Pradeep, “An overview of the routing techniques employed in wireless sensor network,” in 2020 International Conference on Communication and Signal Processing (ICCSP), pp. 332–336, Chennai, India, 2020.

[16] T. Gao, J. Y. Song, J. Zou, J. Ding, D. Wang, and R. Jin, “An overview of performance trade-off mechanisms in routing protocol for green wireless sensor networks,” Wirel. Networks, vol. 22, no. 1, pp. 135–157, 2016.

[17] M. Kumar, R. Tripathi, and S. Tiwari, “Qos guarantee towards reliability and timeliness in industrial wireless sensor networks,” Multimedia Tools and Applications, vol. 77, no. 4, pp. 4491–4508, 2018.

[18] V. Ukani and D. Thacker, “Qos aware geographic routing protocol for multimedia transmission in wireless sensor network,” in 2015 5th Nirma University International Conference on Engineering (NUiCONE), pp. 1–6, Ahmedabad, India, 2015.

[19] T. Qiu, Y. Lv, F. Xia, N. Chen, J. Wan, and A. Tolba, “ERGID: an efficient routing protocol for emergency response internet of things,” Journal of Network and Computer Applications, vol. 72, pp. 104–112, 2016.

[20] C. Gomez, A. Boix, and J. Paradells, “Impact of lqi-based routing metrics on the performance of a one-to-one routing protocol for IEEE 802.15.4 multihop networks,” EURASIP Journal on Wireless Communications and Networking, vol. 2010, no. 1, 2010.

[21] D. de Oliveira Cunha, O. C. M. B. Duarte, and G. Pujolle, “An enhanced routing metric for fading wireless channels,” in WCNC 2008, IEEE Wireless Communications & Networking Conference, pp. 2723–2728, Las Vegas, Nevada, USA, 2008.

[22] F. Entezami, M. Tunicliffe, and C. Politis, “Find the weakest link: statistical analysis on wireless sensor network link-quality metrics,” IEEE Vehicular Technology Magazine, vol. 9, no. 3, pp. 28–38, 2014.

[23] W. Jiang and H. D. Schotten, “A comparison of wireless channel predictors: artificial intelligence versus Kalman filter,” in ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1–6, Shanghai, China, 2019.

[24] F. Qin, X. Dai, and J. E. Mitchell, “Effective-snr estimation for wireless sensor network using Kalman filter,” Ad Hoc Networks, vol. 11, no. 3, pp. 944–958, 2013.