Design of Link-Quality-prediction-based Software Defined Wireless Sensor Networks

Liao Wenxing  
Information Engineering College  
Shaoguan University  
Shaoguan 512000, P.R.China  
Email: liaowenxing@sgu.edu.cn

Shi Xiaofei  
Macau University of Science and Technology  
Macau, P.R.China  
Email: 604767519@qq.com

Abstract—In wireless multi-hop networks, the instability of the wireless links leads to unstable networking. Even in the new designed Software Defined Wireless Sensor Networks (SDWSN), the similar problems exist. To further improve the stability of SDWSN, we introduce a Link Quality (LQ) prediction model into the SDWSN architecture. The prediction model is used to improve the stability between neighbor nodes, and thus the stability of wireless multi-hop routes. Simulation results show that the LQ prediction model can make reasonable corrections to the reality wireless link model, which can well address the restrictive nature of unstable links. As a result, by reducing the use of unstable wireless links, the stability of the SDWSN network improved.

Keywords: Software-Defined Wireless Sensor Networks, Link Quality prediction model, Weak links, Network stability

I. INTRODUCTION

With the development of the Internet of Things (IoT), wireless sensor networks, as an important part of the IoT, also play an important role in an increasing number of fields. An important application scenario for wireless sensor networks is environmental monitoring. In such applications, as the scale of the network increases, the use of a single sink node to cover the monitoring area is increasingly unable to meet the application requirements. Therefore, most wireless sensor networks use a wireless multi-hop networking method. Although this networking method can greatly increase the scale of the network, the accumulation of errors in wireless multi-hop transmission makes networking and multi-hop data transmission extremely unstable.

Most wireless multi-hop networks, including wireless sensor networks, use a distributed routing strategy at the network layer. Our previous research [1-2] has demonstrated that, due to the instability of wireless transmission, regardless of whether active or passive routing strategies are used, the probability of routing success and network stability are greatly limited as network scale increases.

In view of this, researchers have tried to introduce the software-defined ideas in wired networks into wireless sensor networks to reduce or weaken the restrictive nature of unstable links in distributed networking. Some typical software-defined wireless sensor network architectures have been proposed [3-9]. In [3], the challenges and design requirements are discussed. There are two major architectures, two-layer architecture[4-5] and three-layer architecture[6]. In our previous study [7], the feasibility as well as the advantages and challenges of software-defined wireless sensor networks (SDWSN) has been analyzed in detail. Beside the design of SDWSN architecture, some researchers focus on the QoS of SDWSN [8] and new technologies (Edge computation [9] for example) in SDWSN. Nevertheless, there are still limitations in the SDWSN architecture that cannot be ignored, mainly in the following two aspects. Firstly, in the centralized controlled networking approach, the distributed neighbor discovery process is inevitably necessary in order to obtain node connection relationships. The existence of weak links still leads to variable neighbor relationships, which leads to frequent topology repair[2]. Secondly, when the controller collects the neighbor information, it still adopts a broadcast method similar to passive routing protocols. Therefore, the weak links may be used, which leads to the failure of topology collection process.

The above problems are mainly caused by the uncertainty of wireless transmission. When two nodes are on their respective coverage edges, the link becomes unstable due to the temporal and spatial contingency of the wireless signals, which affects the stability of the neighbor relationship. In addition, the error accumulation characteristic of wireless multi-hop transmission further worsens this instability, which brings challenges in stable networking in SDWSN.

To address these issues, we propose an LQ prediction model-based SDWSN architecture, and implement the wireless multi-hop networking mechanism under this architecture. The LQ prediction model can predict the link conditions at the next broadcast period from the historical packet reception of wireless links, so that the existence of unstable links can be predicted in advance and the use of unstable links can be avoided during the networking strategy. Therefore, the application of LQ prediction models can reduce the possibility of using unstable links, which helps to improve the stability of wireless connections between neighbors and the reliability of multi-hop transmission.

At present, the researchers have proposed a number of link quality prediction methods, including traditional optimization algorithms, such as (EWMA)[11,24] and Machine learning-based algorithms, such as (fuzzy and svr)[12,13]. The historical values of link quality metrics such as Received Signal Strength Indicator (RSSI), Packet Receive Rate (PRR), Signal-
Noise Ratio(SNR), and Link Quality Indicator(LQI) are used as the input parameters of prediction models. The output parameters mainly include the link quality over a period of time, the package reception for the next period and the received signal strength for the next period. The mainstream link quality prediction models are generally machine learning-based, and can be divided into two types, the off-line [15,18,21] and on-line [14,17,20] learning models. In [14] and [15], the authors propose an on-line learning model (TALENT) and an off-line learning model (4C) respectively. Both link quality prediction models focus on the improvement of packet delivery rate. The authors use the PRR, RSSI, SNR and LQI as the input parameters to train the models. In [16], the authors propose a link quality prediction protocol to deal with the Multi-robot Coordination Tasks. The RSSI, SNR and the neighbor information of nodes are applied in the proposed protocol. Authors in [10] summarize the link quality prediction applications related to power control. In [26], the authors focus on the measurements and characterization of link quality in energy-constrained 802.15.4 wireless sensor networks.

Consider the network model, most of the link prediction models are proposed for the static or quasi-static networks, such as MESH and WSN [14,15,17,20,22,23,24,25]. While for the networks with mobility, (MANET for example) [16,18,21], the additional network dynamic information is necessary for link quality prediction models. Otherwise, authors in [22,25] propose link quality prediction models for indoor and outdoor wireless sensor networks respectively.

A large number of prediction models achieve good prediction results in different application scenarios. However, due to the random property of wireless links, the models are not very accurate for predicting unstable links. Nevertheless, the use of the LQ prediction model is still able to improve the stability of the link well and reduce the instability region between nodes. This will be explained in detail in the following sections.

Due to the stabilizing effect of the LQ prediction model on wireless links, we introduced an LQ prediction mechanism into the original SDWSN architecture [7] and proposed an LQ prediction-based SDWSN architecture. In this network architecture, we use the output results of prediction model to restrict the usage of weak links and improve the stability and reliability of the network.

The main contributions of this paper are as follows:
1) A wireless LQ prediction model is proposed and it is demonstrated that the LQ prediction model can effectively reduce the usage of unstable links during networking process.
2) The design of the network architecture of SDWSN is improved, and the processes of neighbor discovery and topology discovery under the SDWSN architecture are optimized.
3) The LQ prediction model is applied in SDWSN to effectively avoid the dynamic change of neighbors and improve the stability of the network.

The rest of this paper is organized as follow. In section II, we design a machine learning-based wireless LQ prediction model and analyze the usefulness of the model in stabilizing links in detail. In section III, we design an SDWSN architecture based on the link prediction model and refine the design of the protocol flow. In Section IV, we build a simulation platform to verify the improvement of network performance by engaging the LQ prediction model. Finally, we conclude this paper in section V.

II. DESIGN AND ANALYSIS OF WIRELESS LINK PREDICTION MODEL

A. The structure of LQ prediction model

In this paper, we propose a machine learning-based link prediction model. The structure of the proposed model is as shown in Fig 1.

![Fig. 1: The Structure of LQ prediction model](image)

The inputs to the model combine physical parameters with logical parameters. Suppose a HELLO packet is sent periodically between two nodes, by using the physical parameters and the logical parameters from the $1_{st}$ to the $k_{th}$ broadcast period, we can predict the reception of the HELLO packets in the $(k + 1)_{th}$ period. In this paper, the physical parameters use the RSSI value of the HELLO packets and the logical parameters are the reception of the historical HELLO packets.

First, $i = \{RSSI, RECV\}$ is defined as the input parameter of the model and the reception in the $(k + 1)_{th}$ broadcast period is defined as the output of the model. We obtain these parameters based on the packet reception during the previous $k$ broadcast periods between two nodes in a certain link.

$RSSI_i$ indicates the signal strength of the $i_{th}$ HELLO packet and $RECV_i$ indicates the reception of the $i_{th}$ HELLO packet, where $RECV_i = 1$ indicates that the $i_{th}$ HELLO packet was successfully received and $RECV_i = 0$ indicates that the $i_{th}$ HELLO packet was lost. In the prediction model, we use the $RECV$ and $RSSI$ of the previous $k$ HELLO packets as input parameters and the reception of the $(k + 1)_{th}$ HELLO packet as the predicted output.

Then we build a prediction model based on machine learning, mainly using XGboost, neural networks, etc., and train the model with a large number of input samples. The trained model is able to take in the input feature parameters to accurately predict the reception of packets in the next period. Comparing the predicted output with the actual reception in the next period, we verify the performance of the prediction model.
In practice, the trained model is deployed at the data link layer. When a node receives a packet from its neighbor node, it obtains the historical reception of that node and makes a prediction. If the predicted output is received, the packet is handed over to the network layer for processing. Otherwise, the packet is discarded, indicating that it is a packet that should not be received for processing.

B. The impact of the wireless link model on link stability

A simulation platform has been built to simulate the process of data transmission between wireless nodes. Through the simulation platform, we obtain simulation data for different environments and use them for training and testing. Afterwards, the trained model is deployed to the simulation platform to verify the stability of the wireless LQ prediction model for the link. The data transfer process on the simulation platform is shown in Fig 2.

Fig. 2: HELLO packet broadcast process

1) Channel model: Due to the fading nature of wireless transmission, packet loss may occur when data is transmitted between two nodes. Therefore, we use a log-normal shadow fading based wireless channel model on our simulation platform to simulate the wireless transmission environment between two nodes.

Let \( r \) denotes the distance between nodes \( i \) and \( j \), and the packet delivery rate between these two nodes is given by \( P(r) \). The effective communication radius is \( r_0 \). According to the log-normal shadow fading channel model[1], the packet delivery rate \( P(r) \) is given by

\[
p(r) = \frac{1}{2} \cdot \frac{1}{2} \text{erf} \left( \frac{10\alpha}{\sqrt{2\sigma}} \log_{10} \frac{r}{r_0} \right)
\]

where \( \alpha \) is the path loss exponent and \( \sigma \) denotes the standard deviation, \( r_0 \) represents the effective communication radius, which is the distance when the point-to-point packet delivery rate drops to 50%. Then we have

\[
r_0 = 10^{\frac{\beta_{th}}{10}}
\]

\[
\beta_{th} = \log_{10} \left( \frac{p_t}{p_{r,th}} \right) dB
\]

where \( \beta_{th} \) is the signal attenuation threshold and \( p_{r,th} \) is the threshold of received signal strength, \( p_t \) is the sending signal strength. The curve of \( P(r) \) is shown in fig 3. In this paper, we set \( \alpha = 3, \sigma = 4 \) and \( \beta_{th} = 66dB \).

Fig. 3: Delivery rate of two nodes with distance \( x \)

2) Performance of the prediction model: In order to understand the impact of the wireless LQ prediction model on unstable links, we have analyzed the performance of the prediction model in the following aspects.

- Accuracy

We use the node’s historic information of reception for the first K periods as parameters and use common machine learning models such as Regression model, SVM, Random Forest, XGBoost and Neural networks for link prediction. To better measure the performance of the LQ prediction models, we analyze the performance metrics obtained for each model, such as F1 values, ACC and recall. The results are as shown in Table I. Moreover, we performed a large number of simulations at different distances to obtain samples with distances between nodes uniformly distributed in the range of 0 to \( 2r_0 \).

From Table I, we obtain that with suitable input and output parameters, the LQ prediction models based on different machine learning algorithms do not differ significantly and all achieve good prediction accuracy.

In fact, accuracy is of little importance when it comes to wireless link prediction. Chance is inevitable in actual transmission due to the randomness of the transmission process. Even if the distance between the two nodes is very close or far, there will be packet loss or packet reception. As a result, the output of the prediction model is also somewhat stochastic, which in turn leads to less accurate predictions. In practice, even though the prediction model is not very accurate, it has
a great advantage in terms of link stability.

- **Comparison of measured and predicted data**

To study the performance of the prediction model in unstable links, we compare the predicted and actual outputs for the three cases of inter-node distances of $0.8r_0$, $r_0$ and $1.2r_0$ respectively. Fig 4 to 6 show a comparison of the predicted and actual outputs of the nodes over a continuous period of time, with inter-node distances of $0.8r_0$, $r_0$ and $1.2r_0$ respectively.

![Fig. 4: The comparison of predicted and actual outputs (0.8r_0)](image1)

![Fig. 5: The comparison of predicted and actual outputs (r_0)](image2)

![Fig. 6: The comparison of predicted and actual outputs (1.2r_0)](image3)

As can be seen from the figures, if the link between two nodes is unstable, the packets are sent and received in a highly random situation without an LQ prediction model. After using the prediction model, the output of the predictor is relatively stable and does not show dramatic fluctuations.

- **Modification of the channel by the prediction model**

We deployed the prediction model on two static nodes at distance $x$, ran it for a period of time and counted the accuracy of the prediction outputs. The prediction accuracy at different distances $x$ is shown in Fig 7.

![Fig. 7: The prediction accuracy at different distances x](image4)

As can be seen from Fig 7, the accuracy of the predicted output varies with the distance. Prediction accuracy is worst when the distance of two nodes are near $r_0$, and highest when they are very close or far apart. This is because the link between two nodes is the most stable when they are very close or far apart. However, when the distance is around $r_0$, the stochasticity is strong and the output of the predictor is almost random. In fact, machine learning algorithms cannot solve the problem of predicting purely random events. This also illustrates that there is little significance in pursuing accuracy in link prediction applications, so we need to approach link prediction models from a different perspective.

We incorporate the LQ prediction model into the actual link model, using the predicted output as the actual output, to investigate what happens to the link model after the prediction model is applied. A comparison of the variation of delivery rate before and after using the prediction model is shown in Fig 8.

As can be seen from Fig 8, the delivery rate curve becomes steeper with the use of the prediction model. This corresponds
to the fact that we have reduced the effective communication radius of the node, but the stability of the wireless link around the node’s coverage radius has improved considerably. If we define a wireless link with a delivery rate between 30% and 70% as an unstable link, with the use of wireless LQ prediction model, it is clear that the range of unstable areas has been reduced. From previous theoretical studies, we know that the reason for the instability of wireless multi-hop networks is due to the presence of an instability region, the larger this region, the more unstable the network. The use of link prediction is a good way of reducing the presence of unstable links and avoiding the selection of unstable connections as data transmission channels in practical applications, thereby avoiding transmission failures.

In view of the above analysis, the use of a LQ prediction model in the SDWSN network architecture can be a good solution to the situation of unstable links and avoid the use of unstable links that cause network failure and massive packet loss in practical applications.

III. DESIGN OF LINK PREDICTION MODEL BASED SDWSN ARCHITECTURE

In this paper, we propose an SDWSN architecture based on a link prediction model. Improving on the existing SDN-based wireless sensor network architecture, a link prediction model is proposed to improve the stability of the point-to-point link and hence the overall network performance.

A. Restrictive analysis of SDWSN

A software-defined wireless sensor network architecture is proposed in the [7], in which the applicability of software-defined ideas in wireless sensor networks is analyzed in detail and a networking strategy based on the idea of centralized control is designed. In this strategy, the controller node is responsible for the collection of the whole network topology and the generation and distribution of the flow table of the whole network nodes. The ordinary sensor nodes are only responsible for maintaining their own neighbor tables, in order to avoid the limitations brought by the distributed networking algorithm and improve the stability of networking. Despite the use of the idea of centralized control in the SDWSN network architecture, however, as there is no absolutely reliable secure channel for the multi-hop transmission of control information, it is still inevitable that algorithms such as distributed topology discovery are used. A a result, the disadvantages of a distributed large-scale wireless multi-hop network are unavoidable. This is reflected in two main aspects.

Firstly, the distributed neighbor discovery algorithm results in a dynamically changing neighbor table. In wireless multi-hop networks, neighbor discovery is a prerequisite for topology discovery and an essential networking phase. Nodes usually perform neighbor discovery by sending HELLO packets periodically. However, due to the instability of wireless transmission, there is a certain probability of packet loss. A strong connection may experience occasional packet loss, while a weak connection may also experience occasional packet reception. Even if some strategies are used for optimization, such as receiving or losing multiple packets in a row before determining a neighbor change, the neighbor relationship between any two nodes is still in dynamic changes, as has been demonstrated theoretically in [2]. It is this dynamic neighbor change that makes ordinary nodes initiate topology repair frequently, which on one hand increases the networking overhead and on the other hand may bring about network-wide asynchrony.

Secondly, the distributed topology-related algorithms bring about the loss of control information. Since there is no absolutely reliable and secure channel, topology control information has to be transmitted in a wireless multi-hop approach. This process is similar to the routing discovery process in passive routing protocols. Due to the error accumulation nature of wireless multi-hop transmissions and the presence of weak links, the delivery rate of multi-hop transmissions decreases dramatically with increasing network scale. Although the topology collection process in SDWSN does not require specifying the destination node for routing control information in the same way as the route discovery process of passive routing protocols, (or even the return of topology information by all nodes), the process of returning topology information is similar to the process of returning routing responses. In the SDWSN architecture, the control information may also be relayed in the downlink direction using weak links, and therefore also faces a dramatic deterioration in uplink transmission performance. This situation is also present in the flow table rule distribution process, and although this process can be improved by controller-related algorithms, the loss of control information is inevitable due to the presence of weak links in the neighbor discovery process.

The above drawbacks are ultimately due to the inability to guarantee the reliability of each link segment in a wireless multi-hop network and the inability to avoid the existence of weak links. In view of this, we have designed an SDWSN architecture based on LQ prediction to guarantee the reliability of each wireless link segment by link quality prediction, to
minimize the use of unstable links and to improve reliability at the control level. The prediction results are also used at the controller side for flow table generation to improve transmission reliability at the data level.

B. Design of link prediction model based SDWSN

In view of the above flawed analysis, the use of distributed algorithms and wireless multi-hop paths cannot be avoided in SDWSN networks, and this paper introduces link prediction mechanisms into the design of SDWSN architectures.

In the previous section, we designed a machine learning-based model for wireless LQ prediction model. The model uses a large number of historical transmissions between two nodes as input to train the model, and then uses the trained model to predict the next moment of transmissions, as shown in Fig 1. In the previous section, we have verified the superiority of the wireless LQ prediction model in stabilizing links. Therefore, we have introduced an LQ prediction mechanism in the SDWSN architecture to improve the stability of inter-neighbor connectivity and the reliability of multi-hop transmission of control information by reducing the use of weak links. In this way, we address or weaken the restrictive nature of unstable links in topology management and neighbor management.

With the use of the LQ prediction mechanism, we propose a novel SDWSN architecture based on the link prediction model, as shown in Fig 9, which improves on the SDWSN architecture proposed in [7]. Similarly, there are two roles in this network architecture, the controller node and the sensor node, where the controller node is responsible for global network control, including global topology management and flow table rule management, and the sensor node is responsible for local topology management. The control level of SDWSN is implemented by two main processes, namely the neighbor management process and the topology management process.

- Neighbor management process: the network-wide nodes (including the controller) perform local topology discovery, i.e. neighbor discovery, and collect neighbor information periodically.

- Topology management process: Controller-led, responsible for topology discovery, collection and maintenance, etc. There are three main operational phases as follows.

  - Topology discovery phase. After the neighbor information of the sensing nodes has stabilized, the network enters the topology acquisition phase, when the controller node initiates the topology acquisition process to obtain the topology information of the whole network.
  - Flow table deliver phase. The controller generates the optimal flow table for each node through the topology information collected from the whole network and delivers it to the sensor nodes in turn.
  - Topology maintenance phase. All nodes operate stably, and when sensor nodes detect the changes in their neighbor tables, they started a topology repair process to report the changes to the controller.

Inevitably, at the control level of the SDWSN, both neighbor management and topology management use a distributed approach, and the quality of the neighbor relationships obtained in neighbor discovery largely determines the performance of topology discovery. Therefore, we have designed the SDWSN architecture based on an LQ prediction model to maximize the reliability of each link, and thus the reliability of multi-hop transmission of topology control information.

- Neighbor management

Before the controller performs topology collection, the sensor nodes need to perform neighbor discovery and refine their neighbor tables. All nodes, including the controller broadcast HELLORQ packets periodically. After receiving the HELLORQ packet, the nodes determine whether the HELLORQ packet is processed by the LQ prediction model and modify the neighbor list of this node according to the processing result. In order to further ensure the stability of neighbor relationship, the link will be considered unavailable after continuously receive M HELLORQ packets. Additionally, the link will be considered unavailable after continuously last K HELLORQ packets. The neighbor discovery process is shown in algorithm 1.

- Topology management

When the network has been running for some time, the neighbor tables of all nodes are stable. At this time, the controller initiates the topology collection command and broadcasts the TopologyRQ packet for network-wide topology collection. After completing the topology information collection, the controller enters the topology stable state, generates flow table rules for each node and sends them down to each node hop by hop, completing the networking function. During the stable state of the network, once a sensor node detects a neighbor change, the changed local topology is uploaded to the controller. Similarly, during all control commands issued, the sensor node uses a link prediction model to determine whether or not to process the control command. The specific topology management algorithms at the controller side and at the sensor side are as follows.

After the local topology discovery and global topology discovery, the network enters a stable state. At this point, the transmission path of data is completely determined by the flow
Algorithm 1 Process of Neighbor Discovery

Initialization: Set M, K, T and N (N nodes in the network). Set \( \text{recv}=[0,0,...,0], \text{lost}=[0,0,...,0] \) and \( \text{last} \_\text{recv} \_\text{time}=[0,0,...,0] \) (N elements in recv, lost and last_recv_time list). Start a timer \( t \) to periodically broadcast HELLO_RQ. Then start the neighbor discovery process.

Main loop:

1: while 1 do
2:     Wait until an interrupt occur
3:     if A HELLO_RQ packet from node i is received then
4:         Get the historical reception of HELLO_RQ packets of node i
5:         Get the prediction result of current HELLO_RQ packet \( \text{pre}_i \)
6:         if \( \text{pre}_i == \text{True} \) then
7:             \( \text{recv}[i]+=1 \)
8:             \( \text{lost}[i]=0 \)
9:             \( \text{last} \_\text{recv} \_\text{time}[i]=\text{current time} \)
10:        if \( \text{recv}[i]=M \) and i not in neighbor list then
11:            Add node i into neighbor list
12:        end if
13:     else
14:         Drop this HELLO_RQ packet
15:     end if
16:     else if Timer \( t \) runs out then
17:         Broadcast a HELLO_RQ packet
18:         for each \( i \in \text{Nnodes} \) do
19:             if \( \text{current time} - \text{last} \_\text{recv} \_\text{time}[i] + T \geq 0 \) then
20:                 \( \text{lost}[i]+=1 \)
21:             \( \text{recv}[i]=0 \)
22:             if \( \text{lost}[i]=K \) and i in neighbor list then
23:                 Move node i out of the neighbor list
24:             end if
25:         end if
26:     end for
27:     Reset the timer \( t \)
28: end if
29: end while

Algorithm 2 Process of Topology Management at Controller

Initialization: Set timer \( t_1 \). Then start the topology management process.

Main loop:

1: while 1 do
2:     Wait until an interrupt occur
3:     if Timer \( t_1 \) runs out then
4:         Broadcast Topology_RQ packet
5:         Set Timer \( t_2 \)
6:     Wait for Topology_RP packets from sensors
7:     while 1 do
8:         if Receive Topology_RP from node i then
9:             Get the neighbor list of node i
10:            Add node i into global topology
11:        else if Timer \( t_2 \) runs out then
12:            Generate and delivery the flow table
13:        end if
14:     end while
15: end if
16: end while

Algorithm 3 Process of Topology Management at sensors

Initialization: Start the topology management process.

Main loop:

1: while 1 do
2:     Wait until an interrupt occur
3:     if Receive a new Topology_RQ from node i then
4:         if The source of this Topology_RQ in neighbor list then
5:             Get the historical reception of HELLO_RQ packets of node i
6:         Get the prediction result of Topology_RQ packet \( \text{pre}_i \)
7:         if \( \text{pre}_i == \text{True} \) then
8:             Generate a Topology_RP packet
9:             Send the Topology_RP to the controller
10:        else
11:            Drop this Topology_RQ packet
12:        end if
13:     else
14:         Drop this Topology_RQ packet
15:     end if
16:     else if Receive a duplicate Topology_RQ packet then
17:         Drop this Topology_RQ packet
18: end if
19: end while

Table rules, and all nodes do not need to know the location of the destination node of the data, and complete the forwarding of data according to the flow table, realizing the separation of the control level and data level. Due to the use of the LQ prediction model, the unstable links can be well excluded by the prediction model in the reception of control information at each stage of the network, effectively improving the stability of the local topology and the global topology.

IV. PERFORMANCE ANALYSIS

To verify the performance of the LQ prediction model-based SDWSN architecture, we built a python-based multi-threaded distributed SDWSN simulation platform.

The simulation parameters are shown in Table II.
### TABLE II: Introductions of some important symbols

| Parameter            | Value                        |
|----------------------|------------------------------|
| Path loss exponent $\alpha$ | 3                            |
| Standard deviation $\sigma$       | 4                            |
| Signal attenuation threshold $\beta_{th}$ | 66dB                          |
| Effective communication radius $r_0$ | 150m                         |
| $M$                  | 1/2/3                        |
| $K$                  | 1/2/3                        |
| Node density $\rho$                      | $(4 - 28) \times 10^{-5}$ nodes/m$^2$ |
| Network size         | 500*500m$^2$                 |

The nodes are evenly distributed in the simulation area and the simulation platform assigns each node an IP address, 127.0.0.ID, where ID is the number of the node. At the same time, the simulation platform assigns two threads to each node for neighbor management and topology management respectively. The controller node is located at the center of the simulation area and is responsible for network-wide control. The distribution of nodes is shown in Fig 10.

![Distribution of nodes](image1)

**Fig. 10: Distribution of nodes**

To verify the change in network performance after the use of the LQ prediction model, we deploy a trained LQ prediction model designed in Section II for each node. When a node receives a packet from another node, the historical acceptance of that node is used as input to the model for link quality prediction. If the prediction result is 1, the packet is processed further, otherwise the packet is discarded.

**A. Analysis of local topology**

We analyze the local topology performance in two main ways. First, we analyze the link stability time between two given nodes A and B. Second, we analyze the neighbor stability time for a given node A at different node densities.

- **Link stable duration**

  Given a distance $d$ between two nodes A and B, neighbor discovery is performed between the two nodes by broadcasting $HELLO_{RQ}$ packets. When node A receives $M$ consecutive $HELLO_{RQ}$ packets from node B, a connection to node B is established, indicating that link $B \rightarrow A$ exists. When node A does not receive $HELLO_{RQ}$ packets from node B for $K$ consecutive broadcast periods, it disconnects from node B, indicating that link $B \rightarrow A$ does not exist. The converse is also true.

  ![Fig. 11: The link stable durations between node A and B (without prediction)](image2)

As can be seen from Fig 11, when the distance between two nodes is small or large, the link is able to remain stable for a long time. When the distance is around $r_0$, the link becomes unstable. As the values of $M$ and $K$ increase, the stable duration increases.

![Fig. 12: The link stable durations between node A and B (prediction)](image3)

**Fig 11 and Fig 12** show the link stable durations without and with the prediction model respectively. It can be seen from Fig 11 that when the distance between two nodes is small or large, the link is able to remain stable for a long time. When the distance is around $r_0$, the link becomes unstable. As the values of $M$ and $K$ increase, the stable duration increases. Fig 12 shows the link stable durations after using the LQ prediction model. As can be seen from the figure, there is a significant improvement in link stability after using the LQ prediction model. The distance between two nodes is around $0.8r_0$, where the link stability is the worst. Combined with Fig 8, it can be seen that after the use of the LQ prediction...
model, the distance between two nodes with a delivery rate of around 50% is around $0.8r_0$, where the link is the least stable. Nevertheless, the link stability is greatly improved relative to the case where no LQ prediction model is used. Again, the link stable duration increases as the values of $M$ and $K$ increase.

- **Neighbor stability duration**
  For a given node A, the number of potential neighbors of node A increases as the node density of the simulation platform increases. A link change between any segment of neighbors results in a local topology change of node A. We define the time for which the neighbor table of node A remains stable as the neighbor stable duration. Neighbor stability in SDWSN with and without the LQ prediction model is studied separately.

Fig. 13: The neighbor stable durations of node A (prediction)

Fig. 14: The neighbor stable durations of node A (without prediction)

Fig 13 and Fig 14 show the neighbor stable durations for different node densities respectively. Fig 13 shows the case of using the LQ prediction model. From the figures, it can be seen that the neighbor stable duration gradually decreases and stabilizes as the node density increases. Overall, the neighbor stable duration improves significantly with the use of the LQ prediction model compared to the case without the LQ prediction model. And as $M$ and $K$ increase, the neighbor stable duration also increases rapidly. Choosing the right values of $M$ and $K$ and using the LQ prediction model can improve the stability of the network very well.

![Fig. 15: The average number of neighbors of node A](image)

Although the prediction model is able to improve the neighbor stable duration well at the cost of making the effective coverage of the nodes smaller, it also excludes unstable connections at the edge of the coverage and improves the stability of the network. Fig 15 shows the change in the number of neighbors before and after the use of the LQ prediction model. As can be seen from the figure, the number of neighbors of the nodes is relatively small after the use of the LQ prediction model.

**B. Analysis of global topology**

In this section, we analyze the performance of the global topology in two main ways. First, the average hop count of multi-hop routes formed by any two nodes is analyzed. Then the stability of the multi-hop routes between any two nodes is analyzed. Here, we use the cumulative delivery rate of multiple wireless links in a multi-hop route, which is also called the end-to-end delivery rate, to measure the stability of a multi-hop route.

Fig 16 shows the hop count between two nodes with different distance before and after applying the prediction model. As can be seen from the figure, the hop count increases as the distance increases and the increasing trend is faster in the network with the LQ prediction model. This is because with the use of the LQ prediction model, the unstable area of the wireless link is reduced, as shown in Fig 8. At this point, the distance of a single hop is reduced and the hop count between two nodes increases with the same distance.

To investigate the stability of wireless multi-hop routes before and after the use of the LQ prediction model, we analyzed the end-to-end delivery rate of multi-hop routes. Fig 17 shows the end-to-end delivery rate between two nodes.
In this paper, we delve into the role of wireless link quality prediction models in improving link stability and analyze the suppression effect of prediction models on unstable links. And we propose a software-defined wireless sensor network architecture based on the link quality prediction model, in which a link quality prediction mechanism is incorporated into the software-defined wireless sensor network architecture. We build a Python-based simulation platform for software-defined wireless sensor networks to analyze the effect of the link quality prediction model on the topology stability performance. The validation results of the simulation platform show that the usage of LQ prediction model can well avoid the application of unstable links, reduce the instability of the network and improve the overall performance of the network.

In the further research, we focus on optimization of the wireless link quality prediction model and the design of software-defined wireless sensor network protocol, and try to introduce reinforcement learning to improve the generalizability of the link prediction model.

**CONFLICT OF INTERESTS**

The authors declare that there is no conflict of interest regarding the publication of this article.

**ACKNOWLEDGMENT**

This work is supported by Information Engineering College, Shaoguan University.

**REFERENCES**

[1] Liao, W., et al. The impacts of weak links on routing process in large-scale multi-hop networks. IEEE Access 5(2017): 12125-12134.
[2] Liao, W., et al. The impacts of weak links on topology discovery process in large-scale wireless multi-hop networks. International Journal of Distributed Sensor Networks 16(11)(2020): 1550147720974526.
[3] Kobo H I, Abu-Mahfouz A M, Hancke G P. A survey on software-defined wireless sensor networks: Challenges and design requirements. IEEE access, 5 (2017): 1872-1899.
[4] Luo T, Tan HP and Quek TQS. Sensor OpenFlow: enabling software-defined wireless sensor networks. IEEE Commun Lett 2012; 16(11): 1896-1899.
[5] Olivier F, Carlos G and Florent N. SDN based architecture for clustered WSN. In: Proceedings of the 9th international conference on innovative mobile and internet services in ubiquitous computing (IMIS), Blumenau, Brazil, 8-10 July 2015.
[6] Han ZJ and Ren WL. A novel wireless sensor networks structure based on the SDN. International Journal of Distributed Sensor Networks 2014 ; 10: 874047.
[7] Liao, W., et al. Design of multi-energy-space-based energy-efficient algorithm in novel software-defined wireless sensor networks. International Journal of Distributed Sensor Networks 13(7)(2017).
[8] Letswamotse B B, Malekian R, Chen C Y, et al. Software defined wireless sensor networks (SDWSN): a review on efficient resources, applications and technologies. Journal of Internet Technology, 2018, 19(5): 1303-1313.
[9] Xu F, Ye H, Yang F, et al. Software defined mission-critical wireless sensor network: Architecture and edge offloading strategy. IEEE Access, 2019, 7: 10383-10391.
[10] Lowrance C J , Lauf A P . Link Quality Estimation in Ad Hoc and Mesh Networks: A Survey and Future Directions. Wireless Personal Communications: An International Journal, 2017.
[11] Renner, C., Ernst, S., Weyer, C., Turau, V. Prediction accuracy of link-quality estimators. In Wireless sensor networks (pp. 1-16). Springer(2011).
[12] Lowrance, C. J., Lauf, A. P. A fuzzy-based machine learning model for robot prediction of link quality. In 2016 IEEE symposium series on computational intelligence (SSCI), December 6-9 (2016)(pp.1-8).

[13] Feng, Y., et al. A Link Quality Prediction Method for Wireless Sensor Networks Based on XGBoost. IEEE Access 7(2019): 155229-155241.

[14] Liu, T., Cerpa, A. E. Temporal adaptive link quality prediction with online learning. ACM Transactions on Sensor Networks (TOSN), 10(3)(2014), 46.

[15] Liu, T., Cerpa, A. E. Data-driven link quality prediction using link features. ACM Transactions on Sensor Networks (TOSN), 10(2)(2014), 37.

[16] Kudelski, M., Gambardella, L. M., Di Caro, G. A. A mobility-controlled link quality learning protocol for multi-robot coordination tasks. In 2014 IEEE international conference on robotics and automation (ICRA) (2014) (pp. 5024-5031).

[17] Di Caro, G. A., Kudelski, M., Flushing, E. F., Nagi, J., Ahmed, I., Gambardella, L. M. Online supervised incremental learning of link quality estimates in wireless networks. In 12th annual workshop on ad hoc networking (MED-HOC-NET) (2013) (pp. 133-140).

[18] Medaiyese, O. and A. P. Lauf. Machine Learning Based Adaptive Link Quality Prediction for Robot Network in Dynamic Environment. 2019 IEEE International Symposium on Robotic and Sensors Environments (ROSE) (2019).

[19] Hughes, J. B., et al. A survey of link quality properties related to transmission power control protocols in wireless sensor networks. 2017 23rd International Conference on Automation and Computing (ICAC)(2017).

[20] Marinca, D. and P. Minet. On-line learning and prediction of link quality in wireless sensor networks. 2014 IEEE Global Communications Conference (2014).

[21] Chen, Y., et al. A Link Prediction Method That Can Learn from Network Dynamics. 2014 IEEE International Conference on Data Mining Workshop(2014).

[22] Gomes, E. B., et al. Evaluation of a Link Quality Estimator in an Outdoor WSN Using a Dedicated Node. 2019 IEEE Symposium on Computers and Communications (ISCC)(2019).

[23] Tran, A. T., et al. Link quality estimation in static wireless networks with high traffic load. Journal of Communications and Networks 17(4)(2015): 370-383.

[24] Mi, X., et al. Research on EWMA based link quality evaluation algorithm for WSN. Proceedings of 2011 Cross Strait Quad-Regional Radio Science and Wireless Technology Conference(2011).

[25] Varghese, A., et al. Poster Abstract: The Utility of Wall-Blockage Modeling for Link Quality Prediction in Indoor IoT Deployments. 2020 IEEE/ACM Fifth International Conference on Internet-of-Things Design and Implementation (IoTDI)(2020).

[26] Jesús G. Servín, and Ángel G. Andrade. Measurements and characterization of link quality metrics in energy constrained IEEE 802.15.4. IEEE (2010).