Wireless Communications and Mobile Computing

# Artificial Intelligence (AI) and In-Network Caching Driven Technologies for Smart Cities

Lead Guest Editor: Jianhui Lv
Guest Editors: Yuhui Shi, Xingwei Wang, Lianbo Ma, and Hui Cheng

# Artificial Intelligence (AI) and In-Network Caching Driven Technologies for Smart Cities

Wireless Communications and Mobile Computing

# Artificial Intelligence (AI) and In-Network Caching Driven Technologies for Smart Cities

Lead Guest Editor: Jianhui Lv
Guest Editors: Yuhui Shi, Xingwei Wang, Lianbo Ma, and Hui Cheng

# Chief Editor



## Associate Editors



## Academic Editors

# Contents

**Reinforcement Learning-Based Service-Oriented Dynamic Multipath Routing in SDN**
Kai-Cheng Chiu, Chien-Chang Liu, and Li-Der Chou
Research Article (16 pages), Article ID 1330993, Volume 2022 (2022)

**A Blockchain-Based Auto Insurance Data Sharing Scheme**
Xiaoguang Liu, Hengzhou Yang, Gaoping Li, Hao Dong, and Ziqing Wang
Research Article (11 pages), Article ID 3707906, Volume 2021 (2021)

**A Hybrid Reliable Routing Algorithm Based on LQI and PRR in Industrial Wireless Networks**
Jie Li, Yang Pan, Shijian Ni, and Feng Wang
Research Article (16 pages), Article ID 6039900, Volume 2021 (2021)

**Predicting Customer Turnover Using Recursive Neural Networks**
Abdullah Jafari Chashmi, Vahid Rahmati, Behrouz Rezasoroush, Masumeh Motevalli Alamoti, Mohsen Askari, and Faezeh Heydari Khalili
Research Article (11 pages), Article ID 6623052, Volume 2021 (2021)

**LSEA: Software-Defined Networking-Based QoS-Aware Routing Mechanism for Live-Soccer Event Applications in Smart Cities**
Yingcheng Zhang and Gang Zhao
Research Article (8 pages), Article ID 8829868, Volume 2020 (2020)

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 1330993, 16 pages
https://doi.org/10.1155/2022/1330993

*Research Article*

# Reinforcement Learning-Based Service-Oriented Dynamic Multipath Routing in SDN

**Kai-Cheng Chiu, Chien-Chang Liu, and Li-Der Chou**

*Department of Computer Science and Information Engineering, National Central University, Taoyuan 32001, Taiwan*

Correspondence should be addressed to Li-Der Chou; cld@csie.ncu.edu.tw

Received 14 July 2021; Revised 8 December 2021; Accepted 14 December 2021; Published 31 January 2022

Academic Editor: Jianhui Lv



The increasing quality and various requirements of network services are guaranteed because of the advancement of the emerging network paradigm, software-defined networking (SDN), and benefits from the centralized and software-defined architecture. The SDN not only facilitates the configuration of the network policies for traffic engineering but also brings convenience for network state obtainment. The traffic of numerous services is transmitted within a network, whereas each service may demand different network metrics, such as low latency or low packet loss rate. Corresponding quality of service policies must be enforced to meet the requirements of different services, and the balance of link utilization is also indispensable. In this research, Reinforcement Discrete Learning-Based Service-Oriented Multipath Routing (RED-STAR) has been proposed to understand the policy of distributing an optimal path for each service. The RED-STAR takes the network state and service type as input values to dynamically select the path a service must be forwarded. Custom protocols are designed for network state obtainment, and a deep learning-based traffic classification model is also integrated to identify network services. With the differentiated reward scheme for every service type, the reinforcement learning model in RED-STAR gradually achieves high reward values in various scenarios. The experimental results show that RED-STAR can adopt the dynamic network environment, obtaining the highest average reward value of 1.8579 and the lowest average maximum bandwidth utilization of 0.3601 among all path distribution schemes in a real-case scenario.

## 1. Introduction

As the diversity of network services increases, users accordingly demand high quality of service (QoS) [1]. Each service may be pursued with different network metrics, such as less response time for voice over Internet protocol (VoIP) and low packet loss rate for file transmission. A traffic engineering scheme must forward the traffic of specific services to routes, whereas the traffic of various services is being transmitted within a network. Some routes have different attributes within a network; thus, the service traffic must be appropriately routed to the corresponding paths. Meanwhile, the utilization of every link must also be balanced. With the abovementioned issues, the main problem can be formulated as follows: given a set of services and network link states, an optimal path must be assigned for the traffic of each network service to meet its QoS requirements and the link utilization must be balanced as much as possible.

The emerging software-defined networking (SDN) paradigm will be a potential solution to dynamically assign paths and obtain network link states. The SDN with the global view of the network enables the rapid and dynamic deployment of network policies [2, 3], which are widely used in enterprise and wide area networks. The control plane within a traditional network device is separated from the data plane in SDN, and a logically centralized controller comes into being. The controller communicates with the network devices via an open-standard protocol, namely, OpenFlow, and the switches that support OpenFlow are known as OpenFlow Switches (OFSs) [4]. By adding flow rules via OpenFlow to an OFS, the OFS can execute the instructions designated by the controller. The flow rules are

in charge of either modifying the packet header fields or forwarding the traffic, which can be used to carry out policies to meet the QoS requirements [5].

In the environment of SDN to enforce policies, a corresponding traffic engineering algorithm or mechanism can be used in the controller [6, 7]. However, before we distribute the paths for each service, the service traffic must be first identified, which is known as a traffic classification (TC) task. The TC can be approximately categorized into three approaches [8, 9]: (a) port-based, (b) payload-based, and (c) machine learning-based. Traditional port-based methods identify packets by the well-known port numbers [10] assigned by the Internet Assigned Numbers Authority [11], which is an instant TC scheme but suffers from dynamic port number utilization [12, 13]. Payload-based approaches inspect the payload within a packet by predefined patterns, which can handle dynamic port numbers but are weak at processing encrypted traffic [14, 15]. Machine learning-based approaches exploit various algorithms to classify service traffic by taking either statistical information or packet bytes as input values [16–18]. In this research, a deep learning TC model constituted of the autoencoder and 1D convolutional neural network (CAPC) is used in the SDN controller. The data collection and processing methods, model construction and training, and performance evaluation of the TC model are presented in our previous work [19]. This study is the first to integrate the deep learning TC model within a network environment.

After service classification, an identified service can be assigned to an ideal route by the learning-based algorithm. This work aims to support the services with their required QoS and simultaneously balance the traffic load. The Reinforcement Discrete Learning Service-Oriented Multipath Routing (RED-STAR) mechanism is proposed to dynamically distribute routes in a network to every service and to tackle the problem. As a deep reinforcement learning (DRL) [20] method, RED-STAR considers the network metrics, that is, bandwidth utilization, link latency, and packet loss rate, as the environment state. The metrics are periodically measured and updated for the route distribution task. However, errors of the obtained measurements may occasionally occur because of software simulation or hardware defects. A metric regularization scheme is included in this work. The deep neural network (DNN) model in RED-STAR takes the regularized environment attributes as input values and generates the output via its inner neural network (NN) computation. Each output value of the DNN model represents the reward value of a route, also known as an action, and the action with the highest reward value is the best route in DNN's perception. The reward scheme is inconsistent for different genres of services because of the varying degrees of QoS. For example, VoIP services attach great importance to latency; thus, high latency results in a low reward. Text messages concern more on packet loss rate; thus, a high packet loss rate also leads to a low reward. The differentiated reward scheme prompts the RED-STAR to allocate the appropriate path to the corresponding service traffic. In addition, high unbalanced link utilization incurs a low reward to balance the utilization of links. The "discrete" learning in RED-STAR is a slight modification from a typical DRL scheme, which will be further discussed in the following sections.

The major contributions of this research are summarized as follows:

(1) A deep learning TC model is integrated within a network environment to classify the incoming packet encapsulated in packet-in messages, which is an innovative implementation.

(2) Custom protocols for network metrics obtainment are designed, and RED-STAR regularizes the measured metrics to provide the DRL model with stable input data.

(3) The reward scheme considers different QoS requirements of services and load balancing issues, and RED-STAR distributes routes to services relying on the custom reward scheme.

(4) The DRL mechanism is applied in the SDN, takes network metrics and service type as the environment, and considers routes in the network as the action set, which is a novel traffic engineering paradigm.

(5) The experiments are implemented with real service traffic (i.e., PCAP file traffic replayed by Bit-Twist [21]) instead of simulated traffic (e.g., randomized packet payload generated by iPerf [22]). The results show that the proposed method performs better than other route distribution schemes when considering load balancing and QoS requirements.

The remainder of this work is organized as follows. Related work and background are discussed in Section 2. The system architecture is illustrated in Section 3. The system workflow is elaborated in Section 4. The proposed RED-STAR route distribution is detailed in Section 5. Experimental results are demonstrated in Section 6. Finally, Section 7 concludes this work.

## 2. Related Work and Background

In this research, two main issues are targeted to be addressed for route distribution: (a) QoS guarantee of network services and (b) bandwidth utilization, offloading and balancing. Existing works regarding both topics are discussed in the following paragraphs, and the background and applications of DRL will also be investigated.

*2.1. QoS Guarantee of Network Services.* The traditional network architecture cannot thoroughly offer the QoS guarantee of each service, whereas the emergence of SDN enables the flexible flow rule addition and accelerates the deployment of QoS routing policies [23, 24]. The common strategy of QoS is to reserve bandwidth for specific services, which guarantees the least available bandwidth for each service. Oliveira et al. [25] used Resource Reservation Protocol and OpenFlow to set up a dedicated channel between a service requester and a service provider, with a static

threshold of bandwidth to guarantee file transfer time. Tomovic et al. [26] utilized the SDN mechanism to offer priority flow bandwidth guarantees, designed an algorithm for route calculation and bandwidth reservation, and compared the performance with the best-effort and shortest path routing and IntServ. However, the requirements of services are not limited to the minimum bandwidth guarantee but involve maximum latency and packet loss rate tolerance. Links may have different network metrics, wherefore a superior path distribution algorithm for service traffic is required. Tseng et al. [27] proposed a multiobjective genetic algorithm (GA) to dynamically forecast the resource utilization and energy consumption in the cloud data center. The GA forecasts the resource requirement of the next time slot according to the historical data in previous time slots. Li et al. [28] presented a novel service functions (SF) deployment management platform that allows users to dynamically deploy edge computing service applications with the lowest network latency and service deployment costs in edge computing network environments. Tseng et al. [29] proposed a gateway-based edge computing service model to reduce the latency of data transmission and the network bandwidth from and to the cloud. An on-demand computing resource allocation can be achieved by adjusting the task schedule of the edge gateway via lightweight virtualization technology.

*2.2. Bandwidth Utilization Offloading and Balancing.* Apart from QoS guarantees, traffic offloading and balancing are inevitable issues, which often occur in a multipath network environment [30]. The SDN controller with the global view of a network can observe the network state and dynamically formulate a strategy to optimize traffic forwarding. Traffic offloading is essential when congestion occurs, and the increase of throughput is the primary goal. Chiang et al. [31] proposed a traffic distribution method to offload the incoming traffic. They utilized Link Layer Discovery Protocol (LLDP) in finding disjoint paths and Dijkstra in finding the shortest path with minimum hop counts to increase the overall throughput of the multipath network. Yahya et al. [32] pointed out the defect of the current prevalent open shortest path (OSPF) algorithms, which are prone to selecting merely one single best path for traffic forwarding and likely incur traffic congestion. The authors have developed a depth-first search algorithm to select several best paths according to link utilization, and the group action feature of OFS is used to distribute traffic across multiple paths. Despite an uncongested network, traffic balancing is still desirable to prevent future congestion. Challa et al. [33] proposed a CentFlow routing algorithm to enhance the node and link utilization depending on the centrality measures and temporal node degree. Tseng et al. [34] integrated the hypervisor technique with container virtualization and constructed an integrated virtualization (IV) fog platform for deploying industrial applications based on the virtual network function. Tseng et al. [35] addressed the design pattern of the 5G micro operator and proposed a Decision Tree-Based Flow Redirection (DTBFR) mechanism to redirect the traffic flows to neighbor service nodes. The DTBFR mechanism allows different $\mu$Os to share network resources and speed up the development of edge computing in the future.

*2.3. Reinforcement Learning.* A typical reinforcement learning (RL) [36] scenario involves three essential elements: an environment, agent, and action set. The agent is the learning entity that receives the state from the environment in a sequence of discrete times, $t = 0, 1, 2, \ldots$, where $s_t$ is the state obtained at time $t$. After receiving $s_t$, the agent will select an action, $a_t$, to be performed by its policy according to the gained information. The environment of $s_t$ will be influenced by $a_t$, thereby transforming into $s_{t+1}$. A reward value, $r_t$, standing for the score of performing $a_t$ in $s_t$, will accordingly be generated by the environment and given back to the agent. On the basis of $r_t$, the agent will determine the performance of the previous action and tune its inner algorithm, attempting to obtain high values under the following states.

$Q$-learning [37] is a representative RL paradigm that has a $Q$-function to estimate the expected reward value of performing an action under a state (i.e., $Q$ value):

$$Q^{\pi}(s,a) = E_{s'}\left[r + \lambda Q^{\pi}\left(s', a'\right) \mid s,\ a\right], \tag{1}$$

where $Q^{\pi}(s,a)$ represents the $Q$ value of an action. The policy $\pi$ determines the action to be performed, and $r$ is the reward value of performing $a$ under $s$. After the state-action pair ($s$, $a$), a new state $s$' comes out. A discount factor $\lambda$ is multiplied by $Q^{\pi}(s', a')$ to reduce the impact of events over time. The $Q$-function in $Q$-learning is implemented with a $Q$ table, storing the expected reward values of each action under each state. Once the reward is obtained, the $Q$ table updates its stored value as follows:

$$Q(s,a) = Q(s,a) + \alpha\left(r + \gamma \max_{a'} Q\left(s', a'\right) - Q(s,a)\right), \tag{2}$$

where $\max_{a'} Q(s', a')$ is the maximum expected reward under the next state, which is multiplied by an adjustable discount factor $\gamma$. The addition of $r$ and $\gamma \max_{a'} Q(s', a')$ subtracted from the original $Q(s,a)$ indicates the error of the $Q$ value predicted by the $Q$-function. The difference is multiplied by a learning rate $\alpha$ and added to $Q(s,a)$ to update the $Q$-function.

Although traditional RL works well in simple tasks, it cannot handle high input dimensionality and it suffers from slow convergence. Combined with the emerging deep learning, which mitigates the abovementioned problems, DRL has appeared. The DRL has recently been applied in several fields, such as video gaming [20], self-driving systems [38], and even computer networking [39, 40]. Hossain et al. [41] raised the issue of situation-aware management to ensure application-driven QoS and utilized link delay and packet loss rate as QoS metrics. DRL-based intelligent routing decision-making is proposed to optimize routing paths, with delay and loss rate as the observation space and weighted delay and loss rate as the reward scheme. Lin et al.

[42] used an RL adaptive routing in a hierarchical SDN network. The customized reward function is calculated with delay, packet loss rate, and bandwidth multiplied by the corresponding weighted parameters. The parameters are tuneable and configured according to the requirements of services.

There are many RL methods that learn some weights and then employ conventional routing algorithms. Yu et al. [43] proposed a deep deterministic policy gradient routing optimization mechanism (DROM) for SDN to achieve a universal and customizable routing optimization. The DROM simplifies the network operation and maintenance by improving the network performance, such as delay and throughput, with a black-box optimization in continuous time. Sun et al. [44] built an intelligent network control architecture TIDE (time-relevant deep reinforcement learning for routing optimization) to realize the automatic routing strategy in SDN. An intact "collections-decision-adjustment" loop is proposed to perform an intelligent routing control of a transmitting network. Stampa et al. [45] designed a DRL agent that optimizes routing. The DRL agent adapts automatically to current traffic conditions and proposes tailored configurations that attempt to minimize the network delay. Pham et al. [46] exploited a DRL agent with convolutional neural networks in the context of knowledge-defined networking (KDN) to enhance the performance of QoS-aware routing. Guo et al. [47] proposed a DRL-based QoS-aware secure routing protocol (DQSP). While guaranteeing the QoS, the DQSP can extract knowledge from history traffic demands by interacting with the underlying network environment and dynamically optimize the routing policy. Rischke et al. [48] designed a classical tabular RL approach (QR-SDN) that directly represents the routing paths of individual flows in its state-action space. QR-SDN is the first RL SDN routing approach to enable multiple routing paths between a given source (ingress) switch and destination (egress) switch pair while preserving the flow integrity. Ibrar et al. [49] proposed an intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based fog computing Internet of Things (IoT) systems (IHSF). IHSF solves several problems related to task offloading from IoT devices in a multihop hybrid SDN-F network context.

Based on the abovementioned related research, the feature of this study is to propose a reinforcement discrete learning-based service-oriented multipath routing to understand the policy of distributing an optimal path for each service. The RED-STAR takes the network state and service type as input values to dynamically select the path a service must be forwarded. The seven papers related to the motivations and problems to be solved in this study are compared. The comparison table is shown in Table 1. Compared with IHSF or other methods, our proposed method considers the type of traffic and selects the best routing path.

## 3. System Architecture

The overall system is an SDN paradigm, which can be regarded into two parts: data plane and control plane. The data plane is in charge of forwarding the traffic; the control plane is the primary site to deploy the custom modules, which are the core components of the architecture.

The details of both parts are illustrated in Figure 1 as a UML diagram. The data plane is composed of OFSs, forwarding the traffic between the server and the client according to the rules deployed in the flow table. The flow table stores the commands delivered by the controller, and OFSs forward the packets or modify the header fields according to the rules. The OFSs communicate with the control plane via OpenFlow channels, whether flow deployment or statistical report. As for the control plane, several custom modules constitute the controller, which is described as follows.

*3.1. Controller.* The controller object is responsible for maintaining the information of routers, links, and service objects within the network. A router stands for an OFS; a link is a path between two OFSs, and service is the traffic type being transmitted. In addition, the controller periodically requests the network metrics and regularizes and updates the obtained metrics. The controller also periodically reallocates the paths for services and trains the DRL agent to improve the path allocation. Regardless of the out-of-band communication or in-band communication between the switch and the controller, the method proposed in this study is applicable.

*3.2. Router.* A router object is an entity in the control plane representing an OFS. When an OFS is activated and notifies the controller, a router object will be instantiated. A router object collects the information of each port by sending the port request messages and explores the topology by sending LLDP messages. Moreover, whenever a packet unmatched to flow rules is sent to the controller, the router will normalize and classify the packet to a service type by the CAPC deep learning model. A router is also in charge of the communication between OFSs and the control plane, such as flow addition and port statistics requests.

*3.3. Classifier.* The classifier is a component of a router object that normalizes and classifies packets, and the classification result will be returned to the router. The normalization and training process are detailed in our previous work [19].

*3.4. Link.* A link object consists of three network metrics: bandwidth utilization, latency, and packet loss rate. The controller is responsible for maintaining and updating the links, and the metrics are regularized before being updated. The metrics are the main factors and state for the future DRL path distribution.

*3.5. Service.* A service object records the service type and its specific reward calculation policy. After the controller distributes a path for a service, the allocated path (last action) will be recorded. Once the network metrics are obtained, the metrics will be the input values of the reward calculation method to generate the reward for the last action.

Table 1: RL signal comparison between our proposed method and related research methods.

| RL signals | State | Action | Reward |
|---|---|---|---|
| DROM [43] | The traffic matrix (TM) of the network | The weights of links in the network | The network operation and maintenance strategy |
| TIDE [44] | The traffic matrix (TM) of the network | The weights of links in the network | The QoS strategy |
| Stampa et al. [45] | The traffic matrix (TM) of the network | The weights of links in the network | The mean network delay |
| Pham et al. [46] | The traffic matrix (TM) of the network | The weights of links in the network | The mean of QoS metrics/the mean of qualified flows |
| DQSP [47] | The frequency of packet-in message, the occupancy rate of the flow table, and the channel occupancy rate | The weight of the node assigned as the next hop | The node packet loss rate, node forwarding delay, and flow table status |
| QR-SDN [48] | The currently selected path for each flow | Determine the path of flow(s) | The sum of latencies along the current paths of the flows |
| IHSF [49] | The path reliability, delay, bandwidth utilization, and the number of disturbed flows in case of link's failure | Determine the path of flow(s) | The path's reliability level, the minimum number of disturbed flows, maximum bandwidth utilization, and minimum delay |
| Proposed RED-STAR | The service type, current bandwidth utilization, packet loss rate, and the latency of each link | Determine the path of flow(s) | The QoS requirements of services and link utilization balancing |

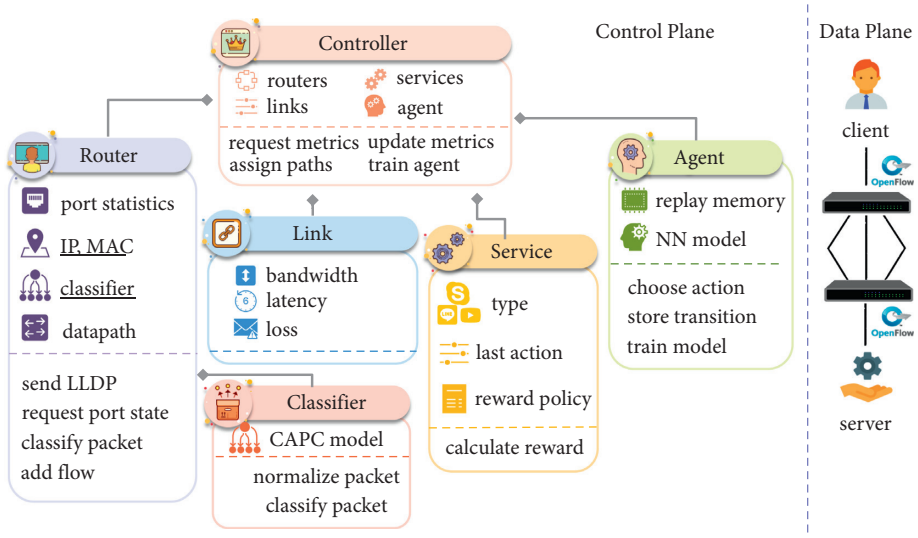


Figure 1: UML diagram of the overall system architecture.

*3.6. Agent.* The agent object belongs to the controller, having an experience replay memory to train the NN model. The NN model is used to select a path for service and trained by the transitions in the replay memory. After path allocation, the previous state, the selected path, reward, and the current state are saved as a transition into the replay memory.

## 4. System Workflow

A few procedures must be accomplished to model the QoS path distribution as an RL problem. This section details each procedure, including network state observation, path distribution, and the reward mechanism.

The typical framing of an RL scenario: an agent takes actions in an environment, which is interpreted as a reward and a representation of the state, which are fed back into the agent. The state of the environment includes the type of service, the current bandwidth utilization, the packet loss rate, and the delay of each link. The description of each procedure is shown in Figure 2, where the observation is to learn the changing environment of the network for the RL agent. The observation includes the service type, current bandwidth utilization, packet loss rate, and the latency of each link. After receiving a state as the input, the agent will select a path for service and add a flow to OFSs. Subsequently, the reward value is generated on the basis of the distribution and service type. The three kinds of data, state, action, and reward, will be stored in the replay memory for agent training.

*4.1. Observation.* A few steps must be completed to form a state, including topology discovery, metric measurement, and metric regularization.

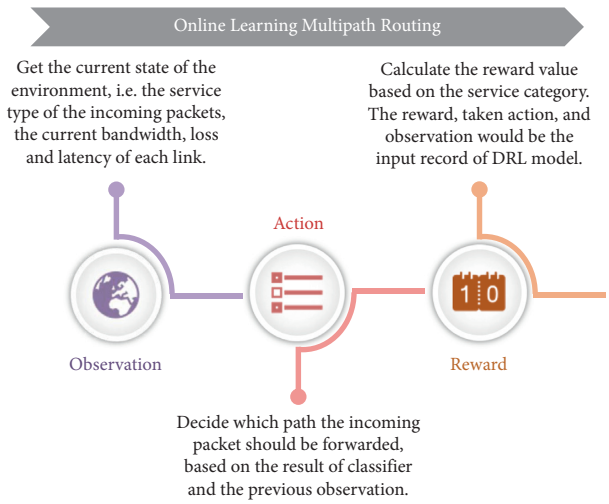


Figure 2: Progress of modeling an RL path distribution problem.

*4.1.1. Link-Layer Topology Discovery.* The task that must be accomplished first is topology construction to offer the paths (action set) for distribution. The LLDP is used to explore the link status of each port. An LLDP packet is crafted with a designated chassis ID, that is, the ID of an OFS, and a port number, thereafter sent out from the port of the OFS. The connected port on the other side will receive the packet, thereby encapsulating the packet in an OpenFlow packet-in message and forwarding back to the controller. The controller can construct the topology of the network. When discovering the link-layer topology, the topology is only a connectivity topology that gives the links between individual network nodes, but does not yet construct end-to-end paths. The entire process of topology discovery is shown in Figure 3, and the notations used in the figure are explained in Table 2.

*4.1.2. Metric Measurement.* The network metrics are measured periodically as the state of the DRL model. The measurement approaches of bandwidth utilization, latency, and packet loss rate are listed in the following order.

*(1) Bandwidth Utilization Measurement.* Every OFS keeps its accumulated transmission byte number up to date. Whenever receiving a port request message, the OFS will answer the port reply to the controller, containing the accumulated transmitted bytes. The controller can thereafter calculate the difference between the previous and the current values, thereby obtaining the bandwidth utilization at this time. The detailed process of the measurement is depicted in Figure 4, and the notations used in the figure can be referred to in Table 3.

*(2) Link Latency Measurement.* A custom protocol is designed to measure the latency. The packet format of the protocol complies with the Ethernet frame; the ether type of which is set as an arbitrary value ($0 \times 8787$). The destination MAC address remains blank, and its source MAC address is filled in with the timestamp at that time. The length of the

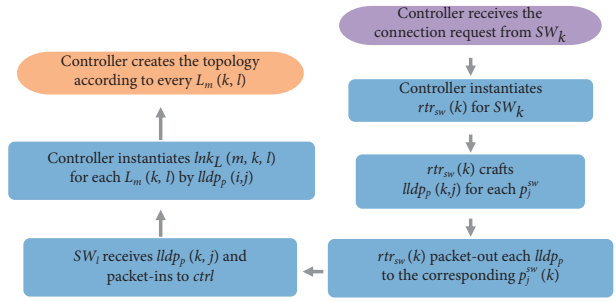


Figure 3: Flowchart of link-layer topology discovery.

custom probing packet is designed as 14 bytes, less than the one used in other research [41, 50]. Whenever an OFS receives a packet with a $0 \times 8787$ ether type, it sends the packet within a packet-in message to the controller. Afterward, the controller removes the timestamp from the packet and calculates the difference between the current and the timestamp in that packet. Finally, the outcome is subtracted from the latency between the OFSs and the controller, and the latency of a link is measured. The overall process of latency measurement is shown in Figure 5, and the notations used in the figure can be referred to in Table 4.

*(3) Packet Loss Rate Measurement.* Similar to the latency measurement, a custom packet format is used for the packet loss rate. The ether type of the packet is set as $0 \times 7878$, and the destination and source MAC addresses remain blank. Initially, the controller sends out a fixed number of probing messages to OFSs. Whenever an OFS receives the packet with a $0 \times 7878$ ether type, it drops the packet immediately. After a fixed period, the OFS delivers the number of $0 \times 7878$ packets it receives from the controller. The controller then calculates the difference between the received number and the original quantity of probing packets sent before. Therefore, the packet loss rate can be calculated. The entire process of packet loss rate measurement is illustrated in Figure 6, and the notations used in the figure can be referred to in Table 5.

*4.1.3. Metric Regularization.* Two aspects must be considered before the utilization of the obtained network metrics. The first aspect indicates that the same value of different metrics has different meanings, for example, latency is presented in milliseconds; bandwidth utilization and packet loss rate are presented in ratio, but 50% bandwidth utilization is definitely better than 50% packet loss rate. The second aspect indicates that some metrics occasionally go wrong.

For the first problem, a mechanism is required to dimension each metric into a similar scale range (0-1), where a larger value is better than a smaller one. In dimensioning a bandwidth utilization value, if the value is originally 0%, then the normalized value will be 1; if the value is originally 100%, then the normalized value will be 0:

$$bw_{\text{lnk}}^{\text{norm}}(m,k,l) = \left(-bw_{\text{lnk}}(m,k,l)\right) + 1, \tag{3}$$

Table 2: Notations for link-layer topology discovery.

| Notation | Description |
|---|---|
| $sw_i$ | The $i^{\text{th}}$ OpenFlow switch |
| $id_{sw}(i)$ | Chassis ID of $sw_i$ |
| $rtr_{sw}(i)$ | Instantiated router object of $sw_i$ in controller |
| $P_j^{sw}(i)$ | The $j^{\text{th}}$ port of $sw_i$ |
| $no_P(j,i)$ | Port number of $P_j^{sw}(i)$ |
| $L_m(k,l)$ | The $m^{\text{th}}$ link between $sw_k$ and $sw_l$ |
| $\text{lnk}_L(m,k,l)$ | Instantiated object of $L_m(k,l)$ in $ctrl$ |
| $lldp_P(i,j)$ | LLDP packet generated for $P_j^{sw}$, composed of $id_{sw}(i)$ and the port number of $P_j^{sw}(i)$ |

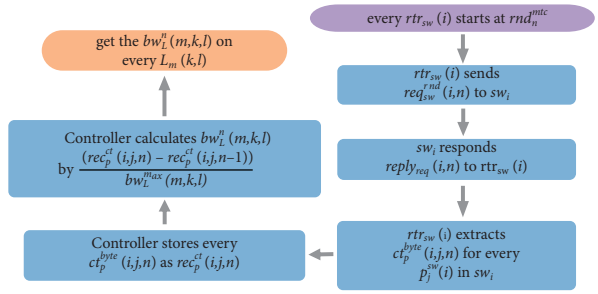


Figure 4: Flowchart of bandwidth utilization measurement.

Table 3: Notations for bandwidth utilization measurement.

| Notation | Description |
|---|---|
| $rnd_n^{mtc}$ | $n^{th}$ round to pull metrics |
| $req_{sw}^{rnd}(i,n)$ | Port statistics request message for $sw_i$ in $rnd_n^{mtc}$ |
| $\text{reply}_{req}(i,n)$ | Port statistics reply message with respect to $req_{sw,rnd}(i,n)$ |
| $ct_P^{\text{byte}}(i,j,n)$ | Total byte count as the sum of $tx$ and $rx$ byte number of $P_j^{sw}(i)$ in $rnd_n^{mtc}$ |
| $rec_P^{ct}(i,j,n)$ | Record saved from $ct_P^{\text{byte}}(i,j,n)$ |
| $bw_L^{\max}(m,k,l)$ | Maximum bandwidth in bytes of $L_m(k,l)$ |
| $bw_L^n(m,k,l)$ | Bandwidth utilization in ratio on $L_m(k,l)$ in $rnd_n^{mtc}$ |

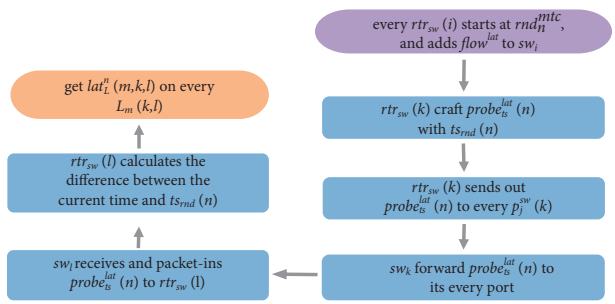


Figure 5: Flowchart of link latency measurement.

Table 4: Notations for link latency measurement.

| Notation | Description |
|---|---|
| $\text{flow}^{\text{lat}}$ | Flow with match field as ether_type = $0x8787$ and action field as packet_in to calculate the latency |
| $ts_{rnd}(n)$ | Timestamp of $rnd_n^{\text{lat}}$ |
| $\text{probe}_{ts}^{\text{lat}}(n)$ | Probe packet crafted with $ts_{rnd}(n)$ for latency measurement |
| $\text{lat}_L^n(m,k,l)$ | Latency in $ms$ on $L_m(k,l)$ in $rnd_n^{mtc}$ |

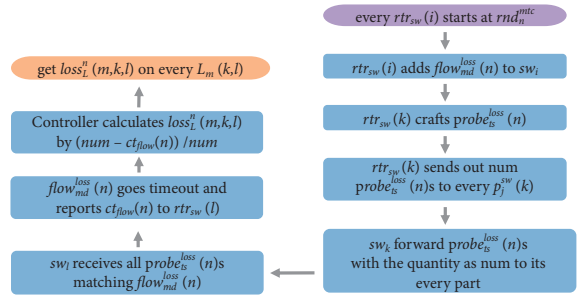


Figure 6: Flowchart of packet loss rate measurement.

Table 5: Notations for packet loss rate measurement.

| Notation | Description |
|---|---|
| $\text{flow}_{rnd}^{\text{loss}}(n)$ | Flow with match field as ether_type = 0$x$7878, action field as none, and hard time out as 1 second |
| $ct_{\text{flow}}(n)$ | Packet count $\text{flow}_{rnd}^{\text{loss}}(n)$ match |
| $\text{probe}_{rnd}^{\text{loss}}(n)$ | Probe packet used for loss measurement at $rnd_n^{mtc}$ number of $\text{probe}_{rnd}^{\text{loss}}(n)$ once to send |
| num | Loss rate in ratio on $L_m(k,l)$ in $rnd_n^{mtc}$ |

where $bw_{\text{ln}k}(m,k,l)$ is the original bandwidth utilization and $bw_{\text{ln}k}^{\text{norm}}(m,k,l)$ is the normalized value. For latency normalization, if the original latency is 0 ms, then the value will be 1; if the original latency is 100 ms, then the value will be 0:

$$\text{lat}_{\text{ln}k}^{\text{norm}}(m,k,l) = -\left(\frac{\text{lat}_{\text{ln}k}(m,k,l)}{100}\right) + 1, \tag{4}$$

where $\text{lat}_{\text{ln}k}(m,k,l)$ is the original latency and $\text{lat}_{\text{ln}k}^{\text{norm}}(m,k,l)$ is the normalized value. Finally, if the original packet loss rate is 0%, then the normalized value will be 1; if the original packet loss rate is 10%, then the normalized value will be 0:

$$\text{loss}_{\text{ln}k}^{\text{norm}}(m,k,l) = (-10) \times \text{loss}_{\text{ln}\,k}(m,k,l) + 1, \tag{5}$$

where $\text{loss}_{\text{ln}k}(m,k,l)$ is the original loss rate and $\text{loss}_{\text{ln}k}^{\text{norm}}(m,k,l)$ denotes the normalized one.

The second problem is solved by the custom mechanism, which is determined by evaluating the difference between the new and the mean value. If the difference is greater than the standard deviation, then it will be determined as an anomaly value and be regularized as the mean value. The dynamic standard deviation can be obtained as follows:

$$\text{Var}(X) = E\left[x^2\right] - E[x]^2, \tag{6}$$

where $\text{Var}(X)$ is the variance and used to calculate the standard deviation.

*4.2. Action Selection.* With the gained input value in the observation, the DRL model can determine a path to forward the traffic of a service. The proposed model is following a complete path approach with a multipath capability. The proposed RED-STAR adopts an *ε-greedy* scheme, in which an $\varepsilon$ probability and 1-$\varepsilon$ probability are found to randomly select an action to be performed and to make the decision on the basis of the calculation result of the DRL model.

Once an OFS receives a packet unmatched to the installed flow, the packet will be sent to the controller within a *packet-in* message. Thereafter, the CAPC model is used to classify the service type that the packet belongs to. Then, a service object is instantiated and added to the service list of the controller. The agent within the controller subsequently allocates a path for each service by the *ε-greedy* approach, thereby training its NN model for good allocation. A forwarding flow of the corresponding service will be added to the OFS. A flow is composed of a matching field and an action field. The matching field is set as the IP address of the IP address and the port number of that service, and the action is set to the forwarding port according to the selected path.

$\varepsilon$ in the *ε-greedy* approach is a variable, which is initially set to a value close to 1. With training, the $\varepsilon$ value gradually decreases (7). Considering that the NN model is not robust at the beginning, the probability of $\varepsilon$ is set to a relatively high value, also known as *exploration*. With time, the decision made by the NN model becomes better and $\varepsilon$ becomes smaller. At present, we can rely more on the NN model to allocate a path for a service, which is known as *exploitation*.

$$\varepsilon \longleftarrow 0.995 \times \varepsilon \,|\, \varepsilon \in [0.01, 1]. \tag{7}$$

*4.3. Reward Scheme.* Two factors are involved in the reward scheme: QoS requirements of services and link utilization balancing. Each factor accounts for the reward value of 1; thus, the maximum reward value is 2.

*4.3.1. QoS Reward.* Each service attaches different importance to the metrics, where a differentiated reward policy is needed. After path allocation for services, the controller will request and receive the metrics for the next turn. Once the controller receives the new metrics, the rewards of services of the last path allocation will be calculated on the basis of their individual policy. A total of 16 applications will be classified into four main categories (Table 6). File transfer services

emphasize the packet quality with less corruption and loss, thereby tolerating high latency. Video streaming and VoIP services are sensitive to link latency, thereby allowing slight packet loss; thus, the latency weight of which must be set higher. Remote control services demand moderate response time and packet loss rate; thus, the weights of the two are set equally. The reward calculation for the four services is formulated as follows:

$$r_{svc}(i) = w_{svc}^{\text{lat}}(i) \times \text{lat} + b_{svc}^{\text{lat}}(i) + w_{svc}^{\text{loss}}(i) \times \text{loss} + b_{svc}^{\text{loss}}(i), \tag{8}$$

where $r_{svc}(i)$ is the reward value of the $i^{\text{th}}$ service. The reward is the sum of the weighted latency and loss rate of the selected path for the service, latency, and loss base value. The latency weight $w_{svc}^{\text{lat}}(i)$ is set higher for latency-sensitive services (streaming and VoIP), and the latency base $b_{svc}^{\text{lat}}(i)$ is set lower. The loss weight $w_{svc}^{\text{loss}}(i)$ is set lower, and the loss base is set higher for the latency-sensitive services. The weight and base values are also set correspondingly on the basis of their QoS requirements. The actual weight and base values for services are depicted in Figure 7. Therefore, the latency and packet loss rate take half of the QoS reward (maximum of 0.5 for each).

*4.3.2. Link Utilization Reward.* An unbalanced path allocation results in a low reward to utilize the bandwidth of the network. The utilization of each link is gathered, and the utilizations of the most and least used link are removed for reward calculation. The large utilization value is subtracted by the small value, and a high difference leads to a low reward value, and vice versa:

$$r_{\text{lnk}}^{bw}(k,l) = -\left(bw_{\text{lnk}}^{\max}(k,l) - bw_{\text{lnk}}^{\min}(k,l)\right) + 1, \tag{9}$$

where $r_{\text{lnk}}^{bw}(k,l)$ is the reward value of utilization balancing. The difference between the maximum and minimum utilization is turned to negative and added by 1. Thereafter, the QoS reward and balancing reward are added as the final reward value of a path distribution:

$$r_{svc}(i) \longleftarrow r_{svc}(i) + r_{\text{lnk}}^{bw}(k,l). \tag{10}$$

## 5. DRL Route Distribution

In a general DRL case, $s_t$ performed with $a_t$ results in $r_t$ and $s_{t+1}$ and a transition $(s_t, a_t, r_t, s_{t+1})$ will be saved in the replay memory. The memory contains several transitions from which the agent arbitrarily selects $n$ transitions to train its NN model. The proposed RED-STAR mechanism is a slight modification (Reinforcement Discrete learning (RED)) of a classic DRL model, that is, deep $Q$-network (DQN). The RED-STAR considers the actual reward $r_t$ affected by $s_t$ and $a_t$, without $s_{t+1}$ (Figure 8).

The main idea of RED is that the route distribution of services does not influence each other directly. For example, the first distribution is targeted at the Skype VoIP service and the next distribution is for LINE VoIP service, whereas the two distributions have no correlation. The state $s_t$ of Skype will not lead to $s_{t+1}$ of LINE. Therefore, a transition stored in the replay memory is constituted of $s_t$, $a_t$, and $r_t$.

Table 6: Targeted applications and their categories.

| File transfer | Video streaming | VoIP | Remote control |
|---|---|---|---|
| FTP | RTP video | LINE VoIP | RDP (Windows) |
| SFTP | RTSP video | Skype VoIP | VMware |
| OneDrive | UDP video | Zoom VoIP | XenServer |
| SCP | YouTube video | Join.me VoIP | NCU cloud |

A typical DRL model involves two NN models: an NN model for action selection and an NN model for the calculation of targeted values. $s_t$ of the transitions randomly selected from the replay memory is fed into the prediction NN model, and the output of which is the action to be performed in this round. $s_{t+1}$ of the transitions is fed into the target NN model, and the output of which is the targeted value to be updated by the prediction model. The output of the prediction model is regarded as the estimated expected reward of performing $a_t$ under $s_t$, whereas the output of the target model multiplied by a discount factor $\gamma$ and added by $r_t$ is considered as the practical expected reward (Figure 9). The prediction NN model trains and updates itself with the practical expected reward as targeted values.

Different from the traditional DRL operation, the output of the NN model in the RED-STAR mechanism stands for the actual reward value $r_t$ for performing $a_t$ under $s_t$, rather than the expected reward. During training, $s_t$ is the input data for the NN model and $r_t$ is set as the targeted value (Figure 10). The model updates its parameters to approximate the targeted value.

The structure of the RED-STAR NN model is depicted in Figure 11, which is a three-layer deep learning structure. The input layer at the top receives 28 features, including the one-hot encoded service type, the network metrics of all routes, and the route allocation state. The output layer contains neurons with the same number as the routes. The value of each neuron represents the reward value for the corresponding path. The mean square error (11) is set as the loss function of the NN model, which is the criterion to determine how "bad" the model is. The *Adam* [51] is a gradient descent method used to update the parameters of the NN model.

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2. \tag{11}$$

To date, the overall contour of the route distribution mechanism has been illustrated. The process of the mechanism can be briefly presented by three procedures: (a) the router objects request their port statistics from the OFSs, thereby obtaining the link states of the topology. The controller then updates the link states received by the router, which is the observation step described in Section 4.1. (b) The agent allocates routes to the services on the basis of its policy, and the controller calculates the reward values of services according to their reward policy. The tasks in this

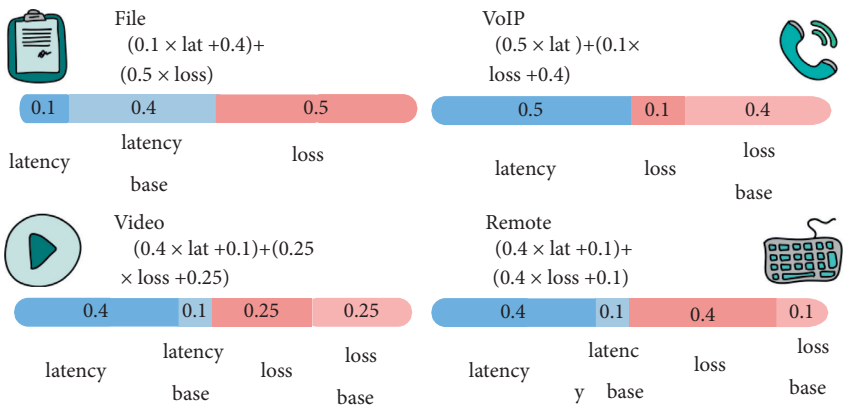


Figure 7: Reward calculation parameters for each service.

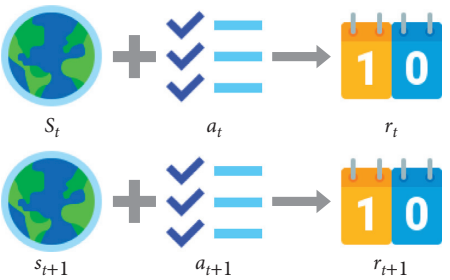


Figure 8: State-action interactions of reinforcement discrete learning.

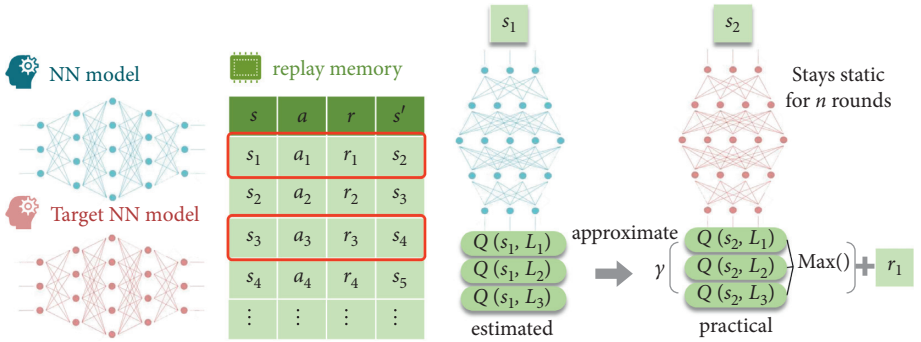


Figure 9: Operation of a traditional DRL model.

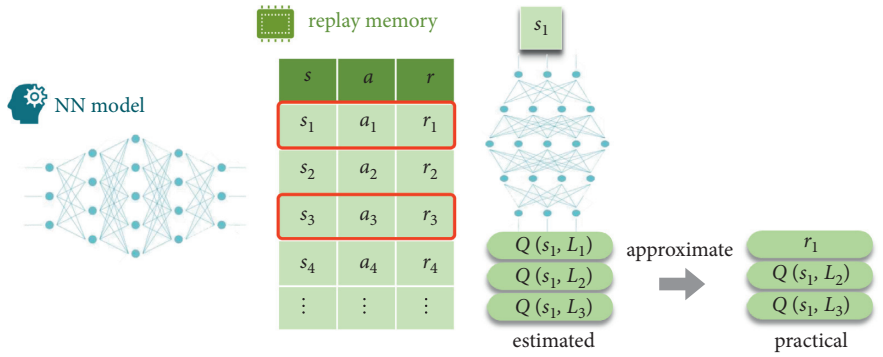


Figure 10: Operation of the RED-STAR model.

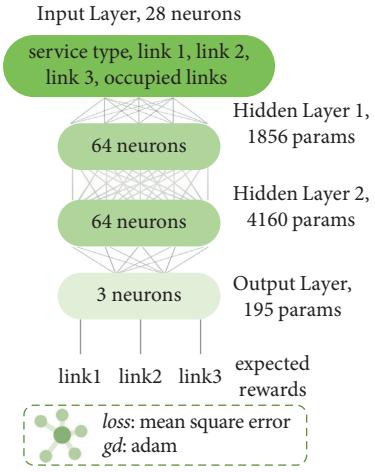


Figure 11: Model structure of the NN model of RED-STAR.

procedure are the action selection and reward obtainment steps in Sections 4.2 and 4.3. (c) The agent trains its NN model on the transitions selected from the replay memory to improve its selection, as the content presented in this section. After training, the overall process returns to step (a) and keeps executing the loop cycle.

## 6. Experiments and Evaluation

Several scenarios are simulated to evaluate the effectiveness and performance of the proposed RED-STAR mechanism. The environment settings, that is, hardware and software specifications, SDN network construction, and link attribute settings, are first introduced. Thereafter, the performance of the RED-STAR mechanism compared with the other two route distribution schemes (i.e., shortest path distribution (SPD) and DQN) in different scenarios is demonstrated.

*6.1. Environment Settings.* The specifications are listed in Table 7. An Ubuntu virtual machine is installed atop a VMware ESXi hypervisor, running Mininet [52] as the network simulator. The Bit-Twist traffic replay toolkit [21] is used for service traffic generation; thus, the traffic in the simulation is the actual PCAP files of certain services. TensorFlow and Keras are used for NN model construction and training.

The network environment topology and configuration are simply set (Figure 12), in which two hosts are in charge of traffic transmission, and the links are featured differently. The delay value and packet loss rate are set proportionally: a higher delay value is configured along with a lower packet loss rate (Link 3), and vice versa (Link 1). The maximum bandwidth values are all set to 100 Mbps. The above-mentioned configuration allows the scheme to determine the route distribution for all services in transmission on the basis of their QoS requirements.

Table 7: Hardware and software specification.

| Hardware/software | Specification |
|---|---|
| CPU | AMD Opteron™ processor 4386 |
| Hypervisor | VMware ESXi 6.7.0 13006603 |
| Operating system | Ubuntu 16.04 LTS |
| Network simulator | Mininet 2.3.0d6 |
| OpenFlow switch simulator | Open vSwitch 2.11.0 |
| SDN controller | Ryu 4.32 |
| Traffic generator | Bit-Twist 2.0 |
| Machine learning engine | TensorFlow 1.14.0 |
| Machine learning toolkits | Keras 2.1.6, scikit-learn 0.22.1 |
| Data preprocessing | Pandas 1.0.0, NumPy 1.16.0 |

*6.2. Reward Scheme.* In this scenario, a LINE VoIP service is transmitted at 10 Mbps in the network. The VoIP is a latency-sensitive and loss-rate-tolerant service, from which we can directly identify that the first link must be the best route for LINE VoIP. A random path distribution scheme is first tested, and its obtained reward values are shown in Figure 13, in which the reward oscillates from 1.6 to 1.8.

The RED-STAR model is prone to arbitrary selection of a route at the beginning based on the $\varepsilon$-*greedy* policy, as $\varepsilon$ gradually declines to rely on the NN model. The reward value gained by RED-STAR is shown in Figure 14(a). The NN model approximately converges after the 200th second and obtains high reward values. Initially, the model selects the third link, causing the expected reward of the third link to grow quicker than the others (Figure 14(b)). After the convergence, the model has been aware of that the first link is more suitable for the LINE VoIP service, thereby fixing its route allocation to select the first link more often and obtain higher rewards.

*6.3. Composition of Different Services.* The traffic of three services is replayed to the network simultaneously to evaluate the performance of each scheme, and the bandwidth consumption of every service is equally set to 10 Mbps in this scenario. The schemes deployed on the controller are in charge of distributing a route for each service. The rewards gained by the three schemes are shown in Figures 15(a)–15(c), where the value of SPD remains the same, and the two learning models obtain higher values with time. The RED-STAR obtains the highest average value for every service.

The DQN model performs worse than RED-STAR, and SPD does not improve with time. The average reward value of the three services of each scheme is presented in Figure 15(d), where RED-STAR converges faster than DQN with the greatest value of 1.8579. Apart from QoS, load balancing is one of the key factors of the reward, which can be separately discussed. In Figure 15(e), the maximum bandwidth utilization of RED-STAR decreases with time, reaching the bottom at around the 500th second, and keeps at a relatively low value of around 0.1. Therefore, RED-STAR has the lowest average utilization (0.1181), indicating that it uses the bandwidth resources in the most effective way.

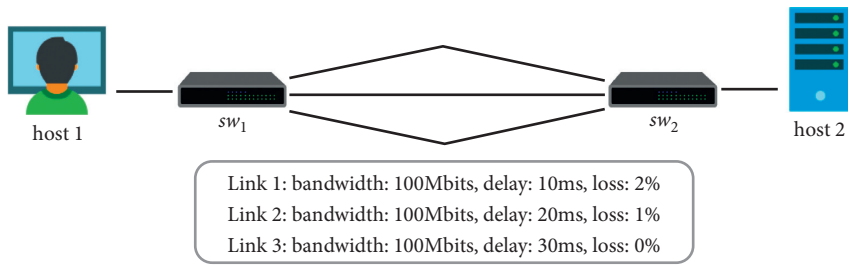


Figure 12: Simulated network topology configuration.

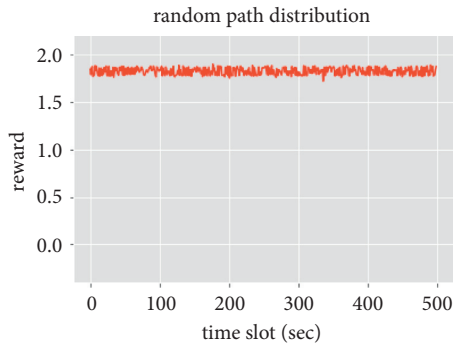


Figure 13: Obtained reward values of a random path distribution scheme.

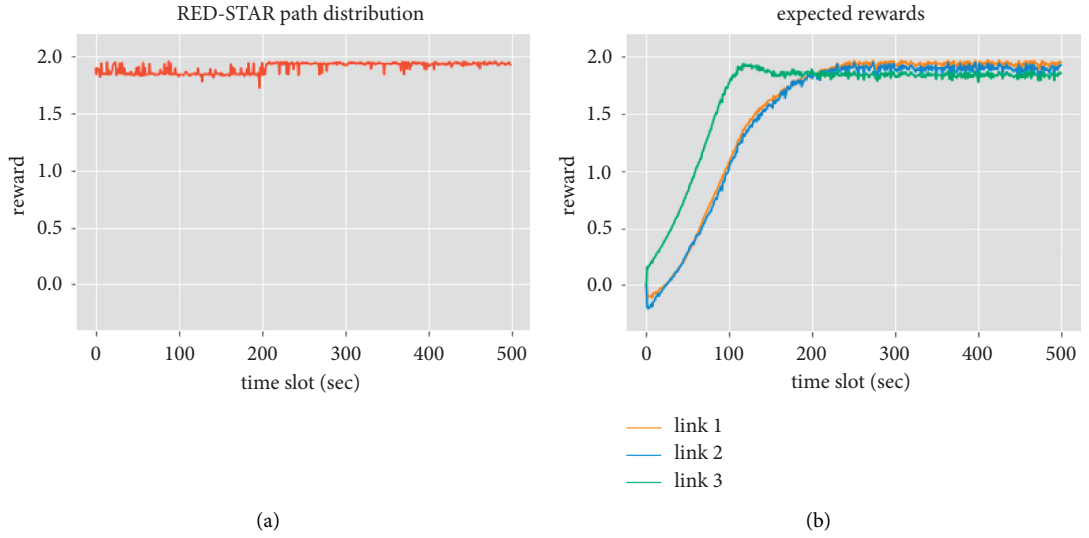


Figure 14: Obtained and expected reward values of the RED-STAR scheme. (a) Actual reward value of RED-STAR and (b) expected reward value of each path of RED-STAR.

*6.4. Composition of Realistic Service Traffic.* In the last scenario, every service has the same bandwidth consumption of 10 Mbps, which cannot reflect the network traffic in the real world. The bandwidth consumption of each service varies according to its category (e.g., VoIP with 1 Mbps, video with 20 Mbps, and file transfer with 40 Mbps). Six services are involved in this case, which is a more complex route distribution task. The reward gained by SPD (Figure 16(a)) remains at a stable reward value; the DQN model approximately converges after the 500th second (Figure 16(b)), obtaining higher values on all services than SPD; the RED-STAR also converges at the 600th second (Figure 16(c)), having average values on three services higher than DQN, and becomes steady after convergence. The average reward of six services in each scheme is presented in Figure 16(d). The SPD scheme keeps at approximately 1.6, whereas the two learning-based models grow steadily from 1.4 to 1.9. The RED-STAR and DQN achieve similar average rewards with 1.8579 and 1.8568, respectively, at the end. Both of them are able to adopt the realistic traffic scenario, of which the models neither have the knowledge of service bandwidth consumption nor take the consumption as the input data. Regarding load balancing, RED-STAR has a lower maximum utilization rate than DQN at the end (Figure 16(e)),

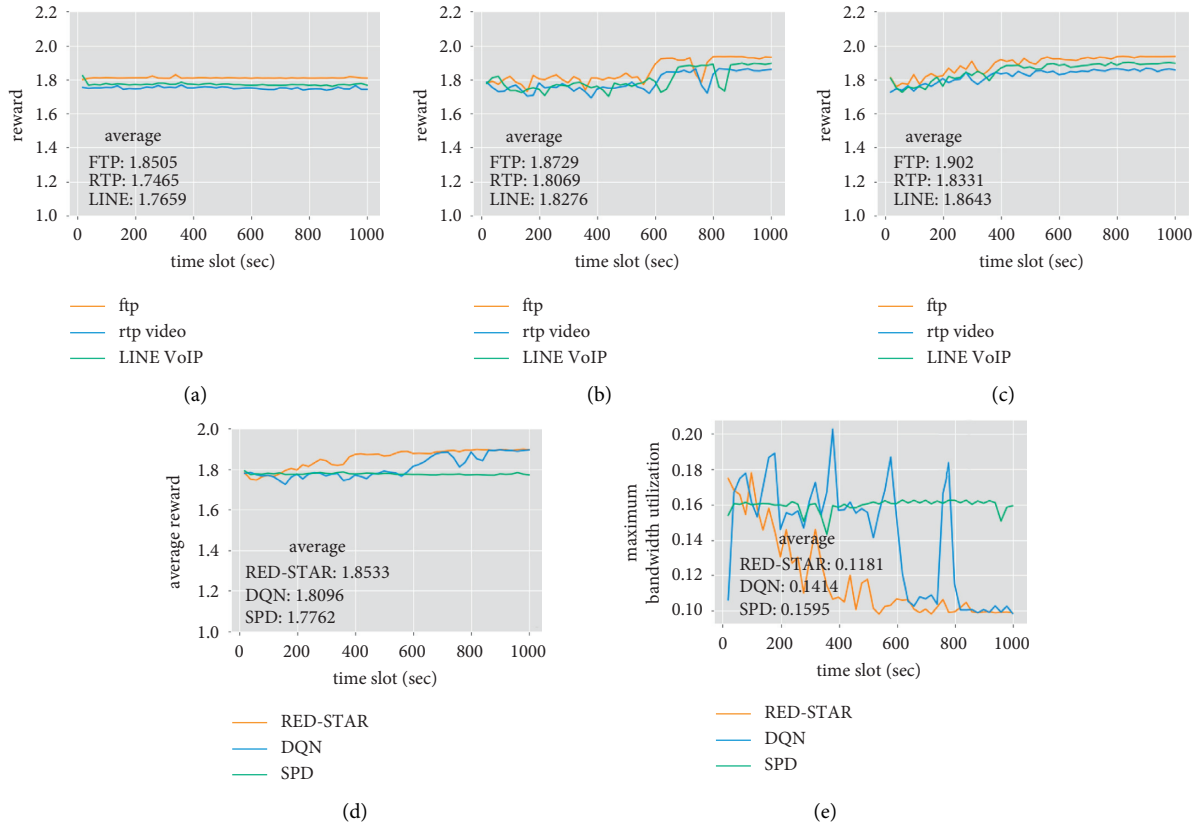


Figure 15: Performance of three path distribution schemes in the three-service scenario. (a) Reward gained by SPD; (b) reward gained by DQN; (c) reward gained by RED-STAR; (d) average reward of three services of each scheme; (e) maximum bandwidth utilization of each scheme.

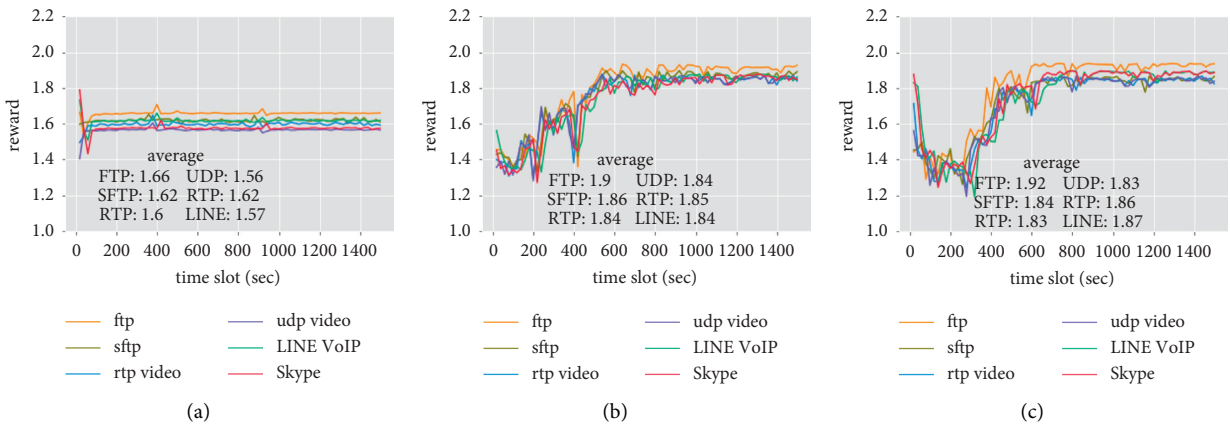


Figure 16: Continued.

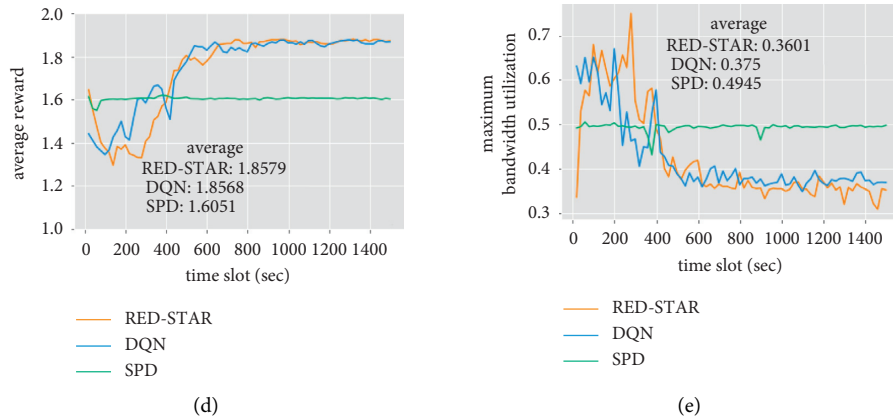


Figure 16: Performance of three path distribution schemes in the realistic six-service scenario (the average values are calculated from the 500th to the 1500th second after convergence). (a) Reward gained by SPD; (b) reward gained by DQN; (c) reward gained by RED-STAR; (d) average reward of three services of each scheme; (e) maximum bandwidth utilization of each scheme.

indicating that DQN gains more reward on the QoS requirements, thereby having a similar average reward as RED-STAR.

## 7. Conclusions

RED-STAR considers the QoS requirements of different services and balances the link utilization with the average reward value of 1.8533, which is greater than the other two schemes, and the lowest average maximum bandwidth utilization of 0.1181 in the three-service traffic scenario. Moreover, in the realistic six-service traffic scenario, RED-STAR still achieves the best average reward of 1.8579 and the lowest average maximum bandwidth utilization of 0.3601 among all schemes.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grants MOST 105-2221-E-008-071-MY3, MOST 108-2221-E-008-033-MY3, and MOST 110-2218-E415-001-MBK.

## References

[1] Y. Chen, T. Farley, and N. Ye, "QoS requirements of network applications on the Internet," *Information—Knowledge—Systems Management*, vol. 4, no. 1, pp. 55–76, 2004.

[2] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: a survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 325–346, 2017.

[3] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: a comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[4] A. Lara, A. Kolasani, and B. Ramamurthy, "Network innovation using OpenFlow: a survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 493–512, 2014.

[5] M. Rezaee and M. H. Yaghmaee Moghaddam, "SDN-based quality of service networking for wide area measurement system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3018–3028, 2020.

[6] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *Proceedings of the IEEE INFOCOM*, pp. 2211–2219, Turin, Italy, April 2013.

[7] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in SDN/OSPF hybrid network," in *Proceedings of the 2014 IEEE 22nd International Conference on Network Protocols*, pp. 563–568, Raleigh, NC, USA, October 2014.

[8] P. Amaral, J. Dinis, P. Pinto, L. Bernardo, J. Tavares, and H. S. Mamede, "Machine learning in software defined networks: data collection and traffic classification," in *Proceedings of the 2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pp. 1–5, Singapore, November 2016.

[9] J. Yan and J. Yuan, "A survey of traffic classification in software defined networks," in *Proceedings of the 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN)*, pp. 200–206, Shenzhen, China, August 2018.

[10] J. Touch, E. Lear, A. Mankin, M. Kojo, K. Ono, and M. Stiemerling, "Service name and transport protocol port number registry," Internet Assigned Numbers Authority (IANA), https://www.iana.org/assignments/service-names-port-numbers.

[11] Iana. Internet Assigned Numbers Authority, Internet Assigned Numbers Authority, https://www.iana.org/.

[12] A. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Proceedings of the International Conference on Passive and Active Network Measurement*, Boston, MA, USA, March 2005.

[13] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," in *Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation*, pp. 179–188, Monterey, CA, USA, September 2006.

[14] M. Finsterbusch, C. Richter, E. Rocha, J.-A. Muller, and K. Hanssgen, "A survey of payload-based traffic classification approaches," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1135–1156, 2014.

[15] D. Sanvito, D. Moro, and A. Capone, "Towards traffic classification offloading to stateful SDN data planes," in *Proceedings of the 2017 IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–4, Bologna, Italy, July 2017.

[16] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Network traffic classifier with convolutional and recurrent neural networks for Internet of Things," *IEEE Access*, vol. 5, pp. 18042–18050, 2017.

[17] P. Wang, F. Ye, X. Chen, and Y. Qian, "Datanet: deep learning based encrypted network traffic classification in SDN home gateway," *IEEE Access*, vol. 6, pp. 55380–55391, 2018.

[18] P. Wang, X. Chen, F. Ye, and Z. Sun, "A survey of techniques for mobile service encrypted traffic classification using deep learning," *IEEE Access*, vol. 7, pp. 54024–54033, 2019.

[19] K.-C. Chiu, C.-C. Liu, and L.-D. Chou, "CAPC: packet-based network service classifier with convolutional autoencoder," *IEEE Access*, vol. 8, pp. 218081–218094, 2020.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, and D. Wierstra, "Playing Atari with deep reinforcement learning," 2013, https://arxiv.org/abs/1312.5602.

[21] A. Heng, "Bit-twist: libpcap-based Ethernet packet generator," 2006, https://bittwist.sourceforge.net/.

[22] C. Hsu and U. Kremer, "IPERF: a framework for automatic construction of performance prediction models," in *Proceedings of the Workshop on Profile and Feedback Directed Compilation*, pp. 1–10, Paris, France, October 1998.

[23] A. Binsahaq, T. R. Sheltami, and K. Salah, "A survey on autonomic provisioning and management of QoS in SDN networks," *IEEE Access*, vol. 7, pp. 73384–73435, 2019.

[24] K. Bouraqia, E. Sabir, M. Sadik, and L. Ladid, "Quality of experience for streaming services: measurements, challenges and insights," *IEEE Access*, vol. 8, pp. 13341–13361, 2020.

[25] A. T. Oliveira, B. J. C. A. Martins, M. F. Moreno, A. B. Vieira, A. T. A. Gomes, and A. Ziviani, "SDN-based architecture for providing QoS to high performance distributed applications," in *Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 602–607, Natal, Brazil, June 2018.

[26] S. Tomovic, N. Prasad, and I. Radusinovic, "SDN control framework for QoS provisioning," in *Proceedings of the 2014 22nd Telecommunications Forum Telfor (TELFOR)*, pp. 111–114, Belgrade, Serbia, November 2014.

[27] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. M. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1688–1699, 2018.

[28] D. C. Li, B.-H. Chen, C.-W. Tseng, and L.-D. Chou, "A novel genetic service function deployment management platform for edge computing," *Mobile Information Systems*, vol. 2020, Article ID 8830294, 22 pages, 2020.

[29] C.-W. Tseng, F.-H. Tseng, Y.-T. Yang, C.-C. Liu, and L.-D. Chou, "Task scheduling for edge computing with agile VNFs on-demand service model toward 5G and beyond," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–13, 2018.

[30] Z. Shu, J. Wan, J. Lin et al., "Traffic engineering in software-defined networking: measurement and management," *IEEE Access*, vol. 4, pp. 3246–3256, 2016.

[31] Y. Chiang, C. Ke, Y. Yu, Y. Chen, and C. Pan, "A multipath transmission scheme for the improvement of throughput over SDN," in *Proceedings of the 2017 International Conference on Applied System Innovation (ICASI)*, pp. 1247–1250, Sapporo, Japan, May 2017.

[32] W. Yahya, A. Basuki, W. Maulana, S. R. Akbar, and A. Bhawiyuga, "Improving end-to-end network throughput using multiple best paths routing in software defined networking," in *Proceedings of the 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE)*, pp. 187–191, Bali, Indonesia, July 2018.

[33] R. Challa, S. Jeon, D. S. Kim, and H. Choo, "CentFlow: centrality-based flow balancing and traffic distribution for higher network utilization," *IEEE Access*, vol. 5, pp. 17045–17058, 2017.

[34] F.-H. Tseng, M.-S. Tsai, C.-W. Tseng, Y.-T. Yang, C.-C. Liu, and L.-D. Chou, "A lightweight autoscaling mechanism for fog computing in industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4529–4537, 2018.

[35] C.-W. Tseng, Y.-K. Huang, F.-H. Tseng, Y.-T. Yang, C.-C. Liu, and L.-D. Chou, "Micro operator design pattern in 5G SDN/NFV network," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3471610, 14 pages, 2018.

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, UK, 1998.

[37] C. J. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[38] J. Duan, S. Eben Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 297–305, 2020.

[39] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[40] M. Latah and L. Toker, "Artificial intelligence enabled software-defined networking: a comprehensive overview," *IET Networks*, vol. 8, no. 2, pp. 79–99, 2019.

[41] M. B. Hossain and J. Wei, "Reinforcement learning-driven QoS-aware intelligent routing for software-defined networks," in *Proceedings of the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 1–5, Ottawa, ON, Canada, November 2019.

[42] S. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: a reinforcement learning approach," in *Proceedings of the 2016 IEEE International Conference on Services Computing (SCC)*, pp. 25–33, San Francisco, CA, USA, June 2016.

[43] C. Yu, J. Lan, Z. Guo, and Y. Hu, "DROM: optimizing the routing in software-defined networks with deep reinforcement learning," *IEEE Access*, vol. 6, no. 18, pp. 54539–64533, 2018.

[44] P. Sun, Y. Hu, J. Lan, L. Tian, and M. Chen, "TIDE: time-relevant deep reinforcement learning for routing optimization," *Future Generation Computer Systems*, vol. 99, pp. 401–409, 2019.

[45] G. Stampa, M. Arias, D. Sanchez-Charles, V. Muntes-Mulero, and A. Cabellos, "A deep-reinforcement learning approach for software-defined networking routing optimization," pp. 1–3, 2017, https://arxiv.org/abs/1709.07080.

[46] Q. T. A. Pham, Y. Hadjadj-Aoul, and A. Outtagarts, "Deep reinforcement learning based QoS-aware routing in knowledge-defined networking," in *Proceedings of the 14th EAI International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness*, pp. 1–13, Ho Chi Minh City, Vietnam, December 2018.

[47] X. Guo, H. Lin, Z. Li, and M. Peng, "Deep-reinforcement-learning-based QoS-aware secure routing for SDN-IoT," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6242–6251, 2020.

[48] J. Rischke, P. Sossalla, H. Salah, F. H. P. Fitzek, and M. Reisslein, "QR-SDN: towards reinforcement learning states, actions, and rewards for direct flow routing in software-defined networks," *IEEE Access*, vol. 8, pp. 174773–174791, 2020.

[49] M. Ibrar, L. Wang, G.-M. Muntean, J. Chen, N. Shah, and A. Akbar, "IHSF: an intelligent solution for improved performance of reliable and time-sensitive flows in hybrid SDN-based FC IoT systems," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3130–3142, 2021.

[50] K. Phemius and M. Bouet, "Monitoring latency with OpenFlow," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, pp. 122–125, Zurich, Switzerland, October 2013.

[51] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations*, pp. 1–41, San Diego, CA, USA, December 2015.

[52] Mininet Team, "Mininet: an instant virtual network on your laptop (or other PC)," 2018, https://mininet.org/.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 3707906, 11 pages
https://doi.org/10.1155/2021/3707906

*Research Article*

# A Blockchain-Based Auto Insurance Data Sharing Scheme

**Xiaoguang Liu**,[1,2,3] **Hengzhou Yang**,[3] **Gaoping Li**,[1,3] **Hao Dong**,[2,3] **and Ziqing Wang**[4]

[1]*School of Mathematics, Southwest Minzu University, Chengdu, Sichuan 610041, China*
[2]*Guangxi Key Laboratory of Cryptography and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China*
[3]*The Key Laboratory for Computer Systems of State Ethnic Affairs Commission, Southwest Minzu University, Chengdu, Sichuan 610041, China*
[4]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China*

Correspondence should be addressed to Xiaoguang Liu; dtcr-gg@163.com

Received 30 July 2021; Accepted 9 November 2021; Published 24 November 2021

Academic Editor: Jianhui Lv



Auto electronic insurance policy and electronic maintenance list record the entire process of auto owners purchasing auto insurance and repairs after accident, respectively. They play a vital role in auto owners' applications for claims and insurance company's judgment on whether to settle the claims. However, the privacy of insurance policy and the "information island" resulting from the nonsharing of data between users make the claim has low efficiency. The notable features of blockchain technology are decentralization and tamper-proof, which can well solve data sharing and privacy protection. This paper proposes a blockchain-based auto insurance data sharing scheme to improve the existing auto insurance claim system. The scheme includes four main bodies: auto owner, insurance company, 4S Shop, and government authority. In the proposed scheme, the data sharing of authorized users is realized through proxy reencryption. Finally, we have analyzed the security and performance of the solution. The analysis results show that the proposed scheme can meet many security features such as user access control and data tamper resistance and has an ideal calculation and communication cost.

## 1. Introduction

With the constant development of society, the auto is everywhere in our life. However, some subjective or objective factors will inevitably cause damage to the auto during the use of the auto. It makes more and more people start to buy insurance for their autos. Now, the auto insurance policy and maintenance list have been digitized with the construction of the smart city. Electronic insurance policy has the advantages of convenient use, low cost, and timeliness and at the same time provides effective evidence for insurance claims. For the time being, these digital documents involve the privacy of auto owners, which shows that privacy protection is a key problem that digital insurance policy must face. The electronic insurance policy and the maintenance list are usually stored and managed by the insurance company and the 4S Shop, respectively. For example, in the process of reviewing claims, the insurance company needs to know the information of insurance purchased by the auto owner and the maintenance case of the auto. The policy information is stored in the insurance company's database, but the insurance company is not clear about the auto's maintenance case. If the insurance company solely relies on the words of the auto owner, it is easy for the auto owner to cheat the insurance and cause losses to the company. Therefore, we propose a scheme based on blockchain to solve the above problem. In the scheme, the insurance company can apply to legally obtain the maintenance records stored in the 4S Shop, and make a quick review to avoid the losses due to the "information island."

*1.1. Related Works.* Since the advent of blockchain [1], people have never stopped researching it. From the initial application in finance to the present, all walks of life are

advocating the "blockchain+" model [2, 3]. The insurance industry also began to join the "blockchain+" model in 2015 and actively try and explore the application of blockchain technology in its own business. As the third largest application scenario in blockchain applications, insurance is usually combined with other fields, such as the insurance and financial fields, auto insurance and maintenance services, and medical insurance and medical fields [4, 5]. Zhao [4] described the current research and application of the insurance industry to blockchain and predicted the trends of "blockchain+insurance." Popovic et al. [5] summarized and gave the issues to be considered when using blockchain technology to solve insurance business problems. Note that in current various industries, the main research about blockchain focuses on information protection, data application, data storage, data sharing, etc., among which data storage and data sharing are the focuses of research [3, 6, 7]. For example, Ekblaw et al. [8] proposed a decentralized record management system that uses blockchain technology to process EMRs. The immutable feature of the blockchain ensures the accuracy of EMRs. However, the scheme did not set a data access strategy, leading to the risk of data leakage. Guo et al. [9] proposed an attribute-based multiauthority signature scheme, which authorizes multiple institutions to manage user attributes on the blockchain. But this scheme is difficult to resist collusion attacks by authorized agencies. Roehrs et al. [10] used the blockchain to build a patient-centric medical architecture model. Unfortunately, the model only integrates the medical data of different medical institutions into one view. These data are still stored on the blockchain, occupying a large amount of storage space on the blockchain. Given this situation, Hua et al. [11] outsourced and stored the data in the cloud after being encrypted, which not only protects patient privacy but also releases storage space of the blockchain. Fu and Fang [12] did further research based on the OPAL/Enigma encryption platform; in NTT services, better encryption algorithms are used to enforce distributed privacy. The scheme uses a trusted certification mechanism to replace proof of work to improve the consensus algorithm of the system. Liu [13] proposed a medical data sharing and protection scheme based on the hospital's private blockchain to improve the electronic health system of the hospital. Particularly, the proposed scheme is implemented by using PBC and OpenSSL libraries.

*1.2. Motivation and Contribution.* At present, since the development of blockchain technology itself is not particularly mature, and the application of blockchain in the insurance field has only been proposed with the construction of the smart city in recent years, the research on "blockchain +insurance" is still in its infancy. The existing research in this area has the following shortcomings [4, 5]: (1) The vast majority of studies only discuss whether it is feasible to apply blockchain to the insurance industry and did not give a specific plan. (2) A small number of studies have given specific application schemes such as data sharing, but there are many disadvantages, such as high cost.

The purpose of this article is to design a blockchain-based auto insurance data sharing model. It can be utilized to help the insurance company store and manage policy data, and share the auto maintenance record information so that a rapid claim settlement is realized and effectively reduces the loss of the insurance company. The solution should be able to protect private information and have an ideal calculation and communication cost. The main contributions of this paper are as follows:

(1) A blockchain-based sharing model of lightweight insurance policy data and auto maintenance records is proposed. By using proxy reencryption technology, this model can realize flexible and secure data sharing

(2) The scheme stores the data of the insurance company and the 4S Shop in their database separately and stores the signatures on the blockchain. This can not only improve the security of the scheme but also reduce all kinds of costs

*1.3. Organization of This Paper.* The rest of this article is structured as follows. First of all, the preliminaries are presented in Section 2. In Section 3, we give a blockchain-based auto insurance data sharing scheme. In Section 4, we analyze the security of the proposed scheme. In Section 5, we analyze the calculation and communication cost of the scheme. Finally, we summarize the proposed scheme in Section 6.

## 2. Preliminaries

*2.1. Blockchain.* The blockchain is mainly to solve the security problems and trust problems generated in the transaction process. It is a distributed database according to a chronological list. Generally, blockchain is divided into the public chain, the consortium chain, and the private chain [14, 15]. In the same blockchain system, all data or data characteristic values will be completely stored by each node. The blockchain structure is shown in Figure 1. A blockchain contains many blocks and the hash value of the previous block is connected to the hash value of the next block. In each block, information such as version number, timestamp, digital signature, and root hash value is stored. The main characteristics of blockchain technology are listed as follows [15]:

(1) Decentralization: there is no special node, and the status of each node is equal. All transactions on the same blockchain are completed by all nodes, and any node can access the data and information on the blockchain. The nodes do not affect each other, and the damage to individual nodes will not have any impact on the system

(2) Tamper resistance: modifying the data will result in a change in the hash value, and the current hash value will affect the hash value of the next node, which causes other nodes to make changes. Therefore, once the data information is written into the block, it

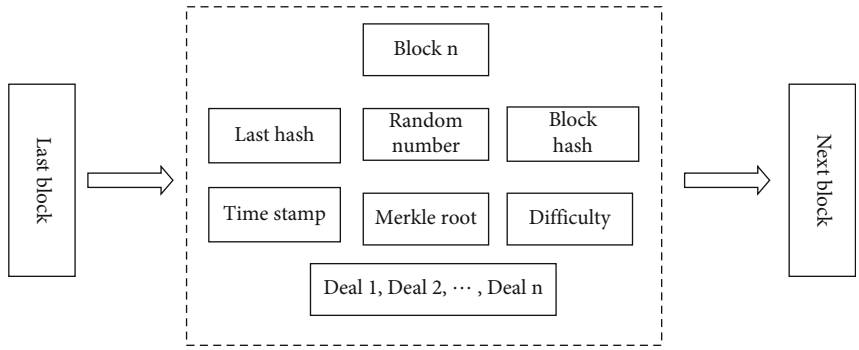


Figure 1: The structure of blockchain.

cannot be changed or canceled unless more than 51% of the nodes are controlled. But in theory, this is very difficult and costly

(3) Openness: in a short period, block transaction information will be copied to all other nodes in the network to realize the synchronization of data in the entire blockchain. Each node can trace the past of both parties to the transaction through its stored information of all transactions

(4) Autonomy: in the system, all nodes can play the role of protector to jointly maintain the entire blockchain system to ensure the reliability and security of information

(5) Anonymity: the identity of each account is encrypted by the algorithm in cryptography. Others can learn the information of this account, but they do not know the identity of the account. Any party on the blockchain will not know any private information of the other party

*2.2. General Network Model.* As shown in Figure 2, the general network model of the blockchain-based auto insurance data sharing scheme is mainly composed of four parties, which are the system manager, insurance company, 4S Shop, and the user who purchases auto insurance. In the model, the system manager plays a vital role in maintaining the normal operation of the entire network. Therefore, this role is usually played by a highly trusted institution such as government departments. When an auto owner needs to purchase auto insurance from an insurance company registered on the blockchain, he/she must first register with an authority to enter the blockchain. Then, the auto owner buys auto insurance from an insurance company. If the policy information is legal, the insurance company stores the policy information in its database and puts the owner's signature on the blockchain for broadcast reception and verification. If the verification is passed, the signature will be stored on the blockchain. When the auto is damaged, the owner sends the auto to a legal 4S Shop for repairs. If the maintenance record information is legal, the 4S Shop will store the maintenance record information in its database and put the owner's signature on the blockchain for broadcast reception and verification. If the verification is passed, it will be stored on the blockchain. Finally, when the insurance company obtains the permission of the auto owner during the claim review, it can check whether the auto repair is reasonable through the proxy reencryption and quickly make compensation.

*2.3. Security Requirements.* Under ideal circumstances, the system should meet the following basic requirements [16]:

(1) Security and privacy protection: insurance policy data and auto maintenance information cannot be illegally used by anyone. The system should be able to resist general malicious attacks and be able to track illegal behaviors

(2) Data access: after being authorized, auto owners can view all their data information, and the insurance company can access auto maintenance information under the authorization of the auto owner

(3) User control: the user can manage all his/her historical data, and no one can obtain the historical data without the user's consent

(4) Unified standards: in the model, all participants should use unified data standards and management schemes, which contribute to data sharing and improve system stability

*2.4. Consensus Mechanism.* A remarkable feature of blockchain technology is that in a decentralized system with decentralized decision-making power and no trust, nodes can reach a consensus on the validity of block data. DPOS is an effective and reliable entrusted proof mechanism [17]. From a certain point of view, it is similar to the board of directors' parliamentary system. All nodes elect 101 representative nodes with equal rights by way of election, and these supernodes will take turns to be responsible for generating a new block. When a node cannot perform its duties, it will lose its accounting rights and be delisted and replaced by a newly elected supernode. The energy consumption of DPOS is lower than that of the POW mechanism, and it is more decentralized than the POS mechanism. It can complete the consensus process faster and improve efficiency.

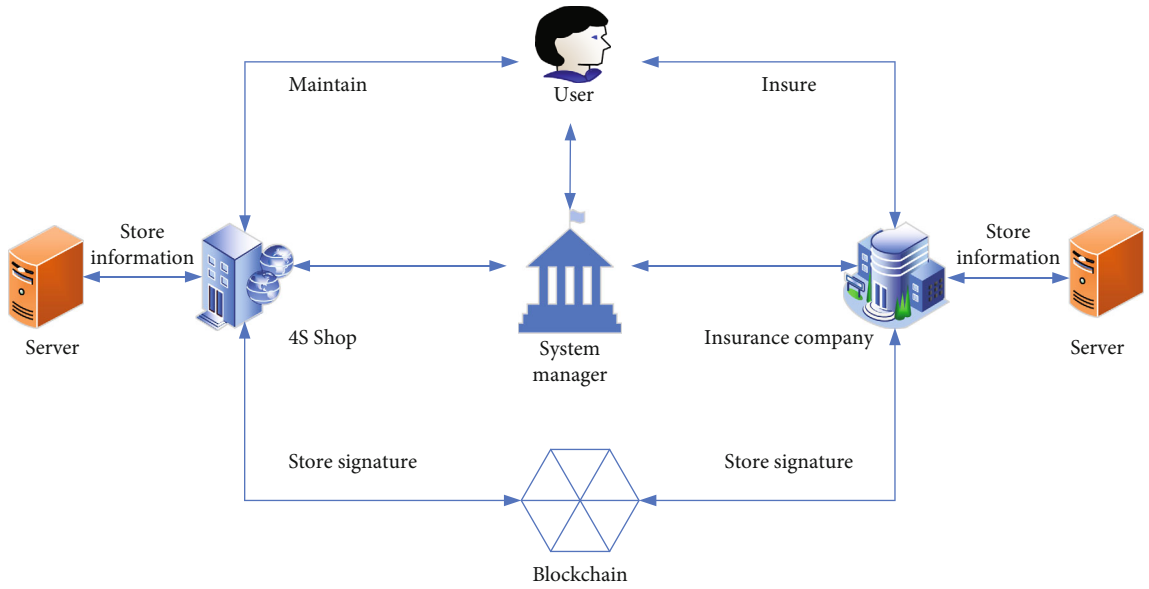


Figure 2: General network model.

*2.5. Bilinear Mapping.* Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic multiplication groups of order $p$, where $p$ is a prime number. If there is a bilinear mapping $e$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ satisfies the following properties, we call $e$ as a bilinear pair [13].

(1) Bilinear: $e(P^a, Q^b) = e(P, Q)^{ab}$, where any $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_p^*$

(2) Nondegeneration: there exists $P, Q \in \mathbb{G}_1$, such that $e(P, Q) \neq 1_{\mathbb{G}_2}$, where $1_{\mathbb{G}_2}$ is the identity element of $\mathbb{G}_2$

(3) Computability: for any $P, Q \in \mathbb{G}_1$, $e(P, Q)$ can be calculated in polynomial time

*2.6. Proxy Reencryption.* Proxy reencryption is an algorithm for reencrypting and decrypting ciphertexts, which was first proposed by Blaze et al. [18] in 1998. In some schemes, a part $A$ entrusts a trusted third party or a semihonest proxy to convert the ciphertext encrypted with its public key into the ciphertext encrypted with the public key of the other party $B$. Then, $B$ can use the private key to decrypt the ciphertext, that is, to realize data sharing. In the whole process, the encrypted data is very safe, and there is no need to disclose $A$'s private key. The specific steps are as follows [19]:

(1) $A$ uses its public key to encrypt the plaintext $m$; that is, $C_A = E_A(m)$, where $m$ is the file that $A$ wants to send to $B$, and $E$ is an asymmetric encryption algorithm

(2) $B$ sends the request to $A$, and then, $A$ (or proxy) generates a conversion key $PK_{A \longrightarrow B}$

(3) $A$ sends $C_A$ and $PK_{A \longrightarrow B}$ to the intermediate proxy

(4) The intermediate proxy converts the ciphertext $C_A$ to $C_B$ through $PK_{A \longrightarrow B}$. At this time, $C_B$ is the ciphertext obtained by encrypting the plaintext $m$ with $B$'s public key. It is worth noting that in this step, the proxy only provides conversion service and cannot obtain plain text

(5) The proxy sends the ciphertext $C_B$ to $B$

(6) $B$ decrypts $C_B$ with its private key to obtain the plain text $m$

## 3. Proposed Auto Insurance Data Sharing Scheme

In this section, we will propose an auto insurance claim scheme based on the alliance blockchain of the insurance company, 4S Shop, policyholder, and the system manager. The property proxy reencryption scheme in [20] is utilized. It has provided a data sharing mechanism for the member of this blockchain. The notations used in this paper are given in Table 1.

As shown in Figure 3, the system manager $SM$, the insurance company $IS_i$, the 4S Shop $4S_j$, and the policyholder $PH_{i;j;n}$ are the four main kinds of participants in the network. $SM$ is the management institution that is a trusted third party and responsible for verifying node identity, generating the master key and system parameters, and verifying the signature of data. Insurance company $IS_i$ and 4S Shop $4S_j$ first register with $SM$. If a person $PH_{i;j;n}$ insures for his/her auto with an insurance company, he/she must register with $SM$ and set his/her public key and private key. If $SM$'s verification has successfully passed, the policy information of $PH_{i;j;n}$ will be stored in the server, and the signatures of $PH_{i;j;n}$ and $IS_i$ will be stored on the blockchain. When the policyholder's auto has an accident, $PH_{i;j;n}$ contacts $IS_i$ to identify the auto and then $IS_i$ checks policy information. If the requirements are met, then it can quickly enter the claim process. $IS_i$ and

Table 1: Notations.

| Notations | Description |
|---|---|
| $SM$ | System manager/government |
| $IS_i$ | $i$th insurance company |
| $4S_j$ | $j$th 4S Shop |
| $PH_{i;j;n}$ | $n$th policyholder |
| $\mathbb{G}_1$ ; $\mathbb{G}_2$ | Cyclic groups |
| $g$ | A generator of the group $\mathbb{G}_1$ |
| $k$ ; $l$ ; $l_1$ | Security parameters |
| $p$ | Large prime number |
| $\|\|$ | String concatenation |
| $H_{(.)}$ | Hash function |
| $ID_{(.)}$ | Identity |
| $F$ | Random function |
| $E_{(.)}$ | Encryption |
| $D_{(.)}$ | Decryption |
| $PK_{(.)}$ | Public key |
| $SK_{(.)}$ | Private key |
| $PID_{(.)}$ | Pseudo-identity |

$PH_{i;j;n}$ first decide which $4S_j$ to repair the auto. Then, $4S_j$ repairs the auto and generates maintenance information. Particularly, the maintenance information of $4S_j$ about $PH_{i;j;n}$ will be stored in $4S_j$'s server, and the signatures of $PH_{i;j;n}$ and $4S_j$ will be stored on the blockchain. If other 4S Shop $4S'_j$ or insurance company $IS'_i$ on this alliance blockchain wants to query the maintenance record information or policy information of $PH_{i;j;n}$, they should apply to the $SM$. If the application is approved, an agent first computes the conversion key. Then, $SM$ generates the ciphertext of the maintenance records or policy information reencrypted by the $4S'_j$'s public key or $IS'_i$'s public key and sends the ciphertext to $4S'_j$ or $IS'_i$. In the following, we will give a detailed introduction of the proposed scheme, which includes six phases, i.e., initialization of system phase, insurance company join phase, 4S Shop join phase, policyholder join phase, signature store phase, and data sharing and search phase.

### 3.1. Initialization of System Phase

(1) Firstly, $SM$ inputs a security parameter $1^k$ and selects the bilinear map $e$ and two multiplicative groups $\mathbb{G}_1$ and $\mathbb{G}_2$, which have the same prime order $p$, and $g$ is a generator of $\mathbb{G}_1$. Secondly, $SM$ chooses three secure hash functions $H_1 : \{0,1\}^* \longrightarrow \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \longrightarrow \{0,1\}^k$, and $H_3 : \mathbb{G}_1 \times \{0,1\}^k \times \{0,1\}^l \longrightarrow \mathbb{Z}_p^*$ and a random function $F : \mathbb{G}_1 \times \mathbb{G}_2 \times \{0,1\}^k \longrightarrow \{0,1\}^{l-l_1} \| \{0,1\}^{l_1}$, where $l$ and $l_1$ are both security parameters. Lastly, $SM$ randomly picks $x \in \mathbb{Z}_p^*$ as the system master key, sets the pubic key $Y = g^x$, selects random elements $g_1, g_2, u, v, d \in \mathbb{G}_1$, and publishes $\{p, e, g, g_1, g_2, u, v, d, Y, H_1, H_2, H_3, F, l, l_1, \mathbb{G}_1, \mathbb{G}_2\}$

(2) The insurance company $IS_i$ randomly picks $x_i \in \mathbb{Z}_p^*$ as its private key and computes public key $PK_i = g^{x_i}$

(3) The 4S Shop $4S_j$ randomly selects $x_j \in \mathbb{Z}_p^*$ as its private key, and the public key is set as $PK_j = g^{x_j}$

(4) The policyholder $PH_{i;j;n}$ randomly chooses $x_n \in \mathbb{Z}_p^*$ as his/her private key and computes the public key $PK_{i;j;n} = g^{x_n}$

### 3.2. Insurance Company Join Phase.

When a new insurance company decides to join the alliance blockchain, it needs to follow those steps combining with $SM$.

(1) $IS_i$ sends its identity $ID_i$ to $SM$

(2) $SM$ verifies its identity; if passes, $SM$ randomly selects $\lambda_i \in \mathbb{Z}_p^*$ and computes $PID_i = E_{SM}(ID_i \oplus \lambda_i) \| \lambda_i$ as $IS_i$'s pseudo-identity

(3) $IS_i$ receives $PID_i$ from $SM$ through a secure channel

### 3.3. 4S Shop Join Phase.

If a new 4S Shop $4S_j$ wants to join the alliance blockchain, it must carry out the following steps combining with $SM$.

(1) $4S_j$ sends its identity $ID_j$ to $SM$

(2) $SM$ verifies its identity; if passes, $SM$ randomly selects $\lambda_j \in \mathbb{Z}_p^*$ and computes $PID_j = E_{SM}(ID_j \oplus \lambda_j \| \lambda_j)$ as $4S_j$'s pseudo-identity

(3) $4S_j$ receives $PID_j$ from $SM$ through a secure channel

### 3.4. Policyholder Join Phase.

If a person buys auto insurance from an insurance company $IS_i$, he/she will become a policyholder $PH_{i;j;n}$. Then, he/she needs to do the following steps, and the index $n$ manifests the policyholder as the $n$th policyholder of $IS_i$.

(1) $PH_{i;j;n}$ sends its identity $ID_{i;j;n}$ to $SM$

(2) $SM$ verifies its identity; if passes, $SM$ randomly selects $\lambda_n \in \mathbb{Z}_p^*$ and computes $PID_{i;j;n} = E_{SM}(ID_{i;j;n} \oplus \lambda_{i;j;n} \| \lambda_{i;j;n})$ as $PH_{i;j;n}$'s pseudo-identity

(3) $PH_{i;j;n}$ receives $PID_{i;j;n}$ from $SM$ through a secure channel

(4) $PH_{i;j;n}$ sends its pseudo-identity $PID_{i;j;n}$ to $IS_i$ and then buys auto insurance in $IS_i$. At the same time, $IS_i$ randomly selects $\delta \in \mathbb{Z}_p^*$ as the evidence for the policyholder and sends $\delta$ to $PH_{i;j;n}$

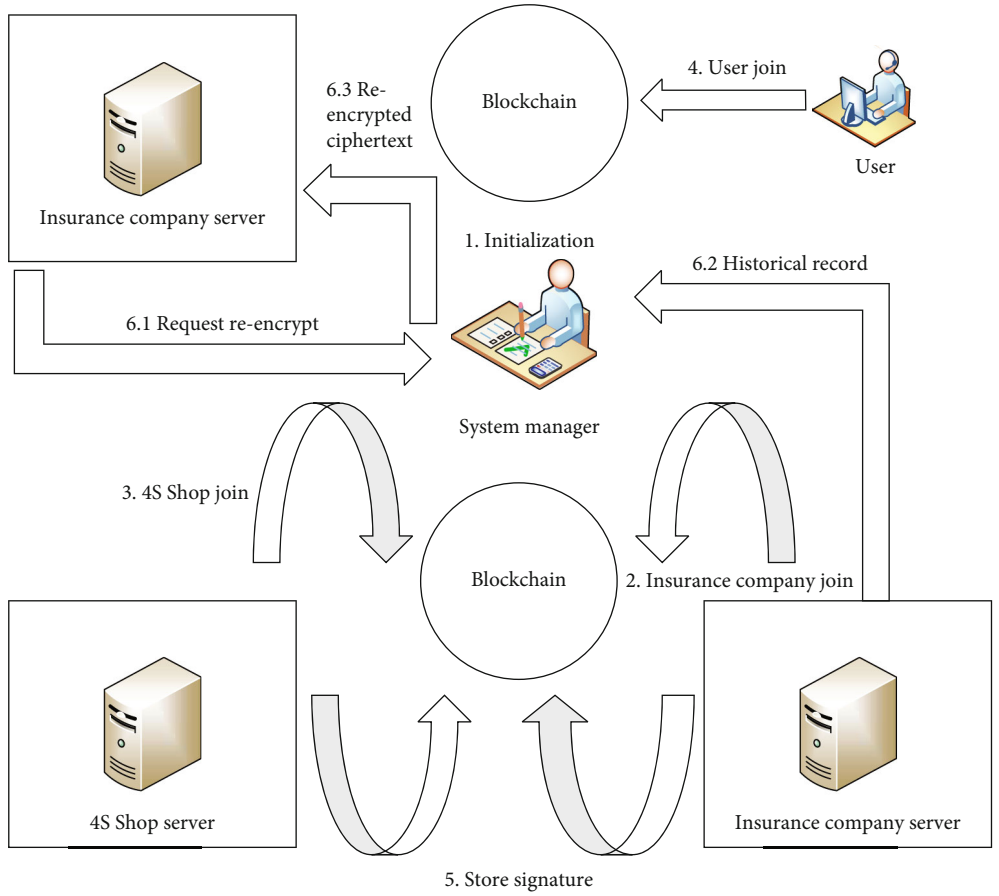


Figure 3: Proposed architecture.

(5) $IS_i$ gives the policy information $m_1$; then, $IS_i$ inputs $PK_{i;j;n}$, $Y$, $m_1$, randomly selects $r \in \mathbb{Z}_p^*$, and computes $\bar{C}_1 = g_1^r$, $\bar{C}_2 = PK_{i,j,n}^r$, $\bar{U} = e(Y, g_2)^r$, $\bar{C}_3 = H_2(U)$, $K = e(g, g)^r$, $\bar{C}_4 = [F(K, \bar{C}_1, \bar{C}_3)]_{l-l_1} \| \{[F(K, \bar{C}_1, \bar{C}_3)]_{l_1} \oplus m_1\}$, $h = H_3(\bar{C}_1, \bar{C}_3, \bar{C}_4)$, and $\bar{C}_5 = (u^h v d)^r$. $IS_i$ stores the ciphertext $\bar{C}_{i;j;n} = (\bar{C}_1, \bar{C}_2, \bar{C}_3, \bar{C}_4, \bar{C}_5)$ in its server and signs the message $m_1$

(6) When policyholder's auto has an accident, $4S_j$ repairs the auto and generates maintenance information $m_2$. Then, $4S_j$ inputs $PK_{i;j;n}$, $Y$ and $m_2$, randomly selects $r \in \mathbb{Z}_p^*$, and computes $C_1 = g_1^r$, $C_2 = PK_{r,j,n}^r$, $U = e(Y, g_2)^r$, $C_3 = H_2(U)$, $K = e(g, g)^r$, $C_4 = [F(K, C_1, C_3)]_{l-l_1} \| \{[F(K, C_1, C_3)]_{l_1} \oplus m_2\}$, $h = H_3(C_1, C_3, C_4)$, and $C_5 = (u^h v d)^r$. $4S_j$ stores the ciphertext $C_{i;j;n} = (C_1, C_2, C_3, C_4, C_5)$ in its server and signs the message $m_2$

*3.5. Signature Store Phase.* In a general DPOS, it needs to elect 101 legitimate participant delegates to record data on the blockchain. In our scheme, the insurance company and 4S Shop are two unrelated departments and have unique professional knowledge. Thus, the general DPOS is not suitable for the alliance blockchain. Because how to elect the delegates is a troublesome problem, and it also needs to take communication and calculation time. In our scheme, we proposed a lightweight and high-efficiency consensus mechanism as we can see in Algorithm 1, and it can be seen as an improvement on DPOS. Each insurance company and 4S Shop can be seen as delegates, who are responsible for broadcasting and recording their own generated data on the blockchain. Due to the high reliability of government agency, it is chosen as the supernode (multiple government agencies can be selected as supernodes to ensure the reliability of the scheme). Moreover, we set up a credit score scheme for the insurance company and 4S Shop to guarantee our mechanism is reliable. *SM* has the right to verify the signature of the insurance company and 4S Shop, if an error signature is found, the credit score of a relevant insurance company or 4S Shop will be reduced. If the credit score is reduced more than three times, it will be expelled from the blockchain. The verification steps of *SM* are listed as follows:

(1) $IS_i$ or $4S_j$ broadcasts the policy data or repair data on the blockchain, respectively

(2) *SM* uses $PK_i$ or $PK_j$ to verify the signature every minute, and then, every twenty legitimate signatures are stored in one new block of the alliance blockchain

1: $IS_i/4S_j$ broadcasts the policy/repair result $C_{i;j;n}$ on the alliance blockchain
2: *SM* verifies the signature
3: if the signature passes the verification
4: the signature is stored in one new block
5: else
6: return FAlSE
7: end if

Algorithm 1: Improved consensus mechanism.

(3) Once the signature is verified, other nodes of the alliance blockchain update their blocks

*3.6. Data Sharing and Search Phase.* The insured repairs the auto in other 4S Shops or purchases auto insurance from other insurance companies; that is, it may need to know the previous insurance policy and auto maintenance records. In this part, we give an example of an insurance company on how to know the auto maintenance information during the claim process. Therefore, after the insurance company obtains $PH_{i;j;n}$'s permission, the algorithm enters $PK_j$ and $PK_i$ and performs the following steps:

(1) $PH_{i;j;n}$ computes $U_1 = g^{r'}$ and $U_2 = g_2^{1/x_n} H_1(Y^{r'})$ and sends $(\delta, U_1, U_2)$ to *SM*, where $r' \in \mathbb{Z}_p^*$ is a random number

(2) *SM* verifies $\delta$; if passes, it sends an extraction instruction about $PH_{i;j;n}$'s policy information to $4S_j$

(3) The agent computes $h = H_3(C_1, C_3, C_4)$; if $e(C_1, PH_{i;j;n}u^h vd) = e(g_1, C_2C_5)$, it computes $C'_2 = C_2^{rk_{i\leftrightarrow n}} = PK_i^r$ and outputs the ciphertext $C'_{i;j;n} = (C_1, C'_2, C_3, C_4, C_5)$ to $IS_i$, where the reencryption key $rk_{i\leftrightarrow n} = x_i/x_n \bmod p$. Otherwise, the stage will be terminated

(4) The server of $4S_j$ sends the encrypted $m_1$ to *SM*

(5) *SM* computes $h = H_3(C_1, C_3, C_4)$; if $e(C_1, PH_{i;j;n}u^h v d) = e(g_1, C_2C_5)$, then it computes $U_\alpha = U_2/H_1(U_1^x)$ and ensures $C_3 = H_2[e(C_2, U_\alpha^x)]$ is true. Otherwise, *SM* outputs $\perp$

(6) $IS_i$ computes $K = e(C'_2, g)^{1/x_i}$; if $[F(K, C_1, C_3)]_{l-l_1} = (C_4)_{l-l_1}$, then $IS_i$ will obtain the ciphertext $m_1 = (C_4)_{l_1} \oplus [F(K, C_1, C_3)]_{l_1}$. Otherwise, the stage will be terminated

## 4. Security Analysis

According to the security requirements given in Section 2.3, this section will analyze the solution from the following security attributes.

(i) Security and privacy: all nodes need to register with *SM* when applying to join the blockchain. *SM* checks whether the identities of the auto owner, insurance company, and 4S Shop are legal. Only nodes with legal identities are allowed to join. After the insurance company or 4S Shop registers with *SM*, *SM* will generate a fake identity for it. When the owner goes to insure or repair the auto, *SM* also calculates a false identity for the owner. In the follow-up process, all nodes use fake identities instead of real identities, and privacy is greatly protected. All transaction information is encrypted by asymmetric encryption, which can effectively prevent unauthorized node access. When the insurance company needs to query the auto owner's maintenance record, the proxy reencryption technology will be used with the owner's consent. When *SM* finishes the confirmation, the agent will convert the relevant maintenance record into a document that the insurance company can decrypt with its private key. In this way, data sharing between different institutions is realized under the premise of ensuring data privacy. Therefore, the solution has good privacy and security

(ii) Data access: this scheme uses proxy reencryption technology to realize data sharing between different institutions. For example, if an insurance company wants to obtain the relevant data stored in the 4S Shop, the insurance company needs to obtain the consent of the applicant. Then, the insurance company will obtain the reencrypted ciphertext and decrypt it with its private key to get the data

(iii) User control: the insurance policy and maintenance records are stored in the respective servers of the insurance company and the 4S Shop. For example, if the insurance company wants to obtain the relevant data of the 4S Shop, the insurance company must first obtain the consent of the applicant. Therefore, the policyholders can control access to data

(iv) Unified standards: in this scheme, we can use unified data standards, such as the keywords of auto damage, which is conducive to data sharing and protection

(v) Tamper resistance: in this solution, the encrypted insurance data and auto maintenance records are stored in the servers of the insurance company and the 4S Shop, respectively, and their signatures are stored on the blockchain. Since the server is not completely trusted, it may tamper with data. One is that the server first colludes with the signer to modify the original data, then resigning the data and replacing the original signature on the blockchain. However, due to the existence of the timestamp, the replaced signature can never be completely consistent with the original signature.

Table 2: Comparison of calculation time (ms).

| | Encrypt | Decrypt | Reencrypt | Redecrypt |
|---|---|---|---|---|
| [22] | $2M + 1E + 2P = 50.87$ | $1M + 3P = 62.50$ | $1P = 20.04$ | $1M + 4P = 82.54$ |
| [23] | $2M + 3P = 64.88$ | $1M + 4P = 82.54$ | $1P = 20.04$ | $1M + 5P = 102.58$ |
| Ours | $3M + 2P = 47.22$ | $1M + 3P = 62.50$ | $2P = 40.08$ | $1M + 5P = 102.58$ |

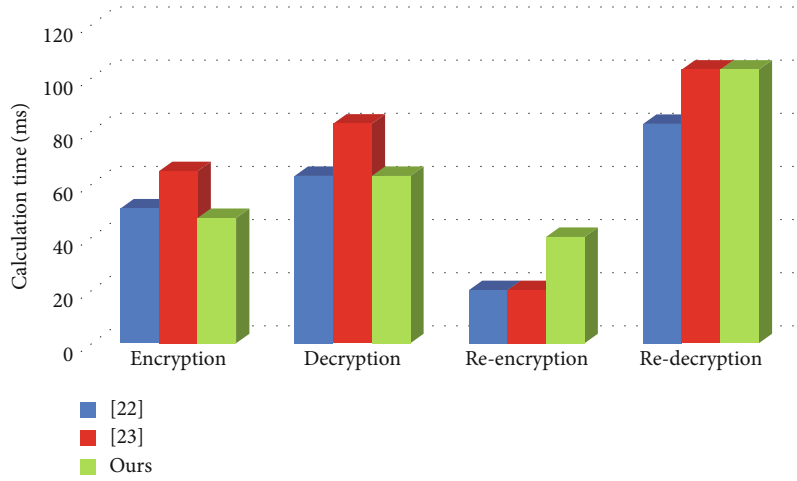


Figure 4: Comparison of calculation time.

Table 3: Comparison of communication cost.

| Schemes | Communication cost (byte) |
|---|---|
| [22] | $8\lvert\mathbb{G}_1\rvert + 4\lvert\mathbb{Z}_p^*\rvert + 1\lvert x\rvert + 1\lvert ID\rvert + 3k + 2l$ |
| [23] | $(n+3)\lvert\mathbb{G}_1\rvert + (n+2)\lvert\mathbb{Z}_p^*\rvert + 1\lvert x\rvert + 4\lvert ID\rvert + 2k + 2l$ |
| Ours | $(n+5)\lvert\mathbb{G}_1\rvert + 5\lvert\mathbb{Z}_p^*\rvert + 1\lvert x\rvert + 2\lvert ID\rvert + 2k + 1l$ |

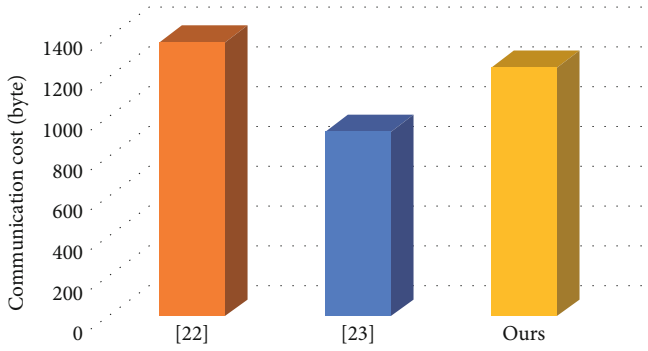


Figure 5: Comparison of communication cost.

Additionally, the signature stored on the blockchain will not be changed due to the immutable feature of the blockchain. The other one is the server forges the signature to modify the data and calculates the private key $SK'$ which is the same as the private key $SK$ of the signature node; that is, $SK' = SK$. However, according to the difficulty of discrete logarithms, this method is not feasible. In summary, the data signature stored on the blockchain is tamper resistance, and it also ensures that the data stored on the insurance company and 4S Shop servers are tamper resistance

(vi) Defend against modification attack: based on the above analysis, we know that it is difficult for the adversary to directly forge a new document to replace the target document, but we need to further consider the adversary's modification attack on the target document. In our scheme, we have two security mechanisms to resist this modification attack. The first is the signature verification mechanism. Under this mechanism, the insurance company, the 4S Shop, and the auto owner need to sign the insurance policy or the maintenance record sheet and then hand it over to $SM$ to verify. Only the successfully verified document is valid. The voucher is based on a secure signature algorithm, and it is also impossible to forge or modify a document [21]. The second is the tamper resistance mechanism based on the blockchain. If the adversary cannot destroy the security of the blockchain, then the adversary cannot destroy the security of the system through a document modification attack. Besides, the use of timestamp also can prevent changes to the data. Therefore, the proposed scheme can resist the modification attack very well

## 5. Performance Analysis

In this section, we will compare the proposed scheme with two similar blockchain-based data sharing solutions [22, 23]. For the convenience of comparison, we use $M$ to represent the multiplication operator, $E$ to represent the power exponent operator in the finite field of prime numbers, and $P$ to represent the bilinear pairing operator. It can be seen from [24] that the cost of a single multiexponentiation is about 1.2 times the cost of single exponentiation. The remaining operators are negligible due to their low calculation time.

In [25], they simulated the user's environment through Windows XP OS on Inter(R) Pentium IV 3.0 GHz processor and 512 MB RAM. At the same time, they simulated the C environment to run on a 32-bit Intel(R) PXA270 624 MHz processor and 128 MB of memory through Windows CE 5.2 OS. The system takes 20.04 ms to execute a bilinear pairing operator $P$, 2.38 ms to execute a multiplication operator $M$, and 5.31 ms to execute a power exponent operator $E$. In this article, we will use the basic test results in [25] to estimate the calculation cost.

According to Table 2 and Figure 4, the proposed solution takes less time in the encryption and decryption stages than the other two solutions. The time in the reencryption and redecryption stages is higher than the other two solutions, and the longest is 102.58 ms. But here the caveat is that the step of reencryption is completed by the agent, and its calculation ability is usually sufficient, so the extra time spent in our solution can be accepted. Particularly, the total time cost is decreased by 7% compared with the scheme in [23].

For the communication cost, the auto owner and the insurance company, the auto owner and the 4S Shop, and the insurance company and the 4S Shop in the three stages of data broadcasting, data verification, and data access are considered. In the proposed scheme, the auto owner $PH_{i;j;n}$ needs to send $(U_1, U_2)$ to $SM$, where $U_1$ and $U_2$ are the elements of $\mathbb{G}_1$. If the insurance company wants to query the owner's maintenance records, the owner $PH_{i;j;n}$ will send the private key $x_n$ to the $SM$, where $x_n$ is the element of $\mathbb{Z}_p^*$. For insurance company and 4S Shop, it also needs to send the private key $x_i$ or $x_j$ to $SM$ and receive the required history records, where $x_i$ and $x_j$ are also elements of $\mathbb{Z}_p^*$, and the ciphertext of the historical records is $C_{i;j;n}$. Also, insurance company and 4S Shop are responsible for broadcasting ciphertexts $C_{i;j;n} = (C_1, C_2, C_3, C_4, C_5)$, block ID, user pseudo-identities, public keys, and signatures on the blockchain. $C_1$, $C_2$, and $C_5$ are elements of $\mathbb{G}_1$, $C_3$ is an element of size $k$, and $C_4$ is an element of size $l$; the user's pseudo-identity is an element of the general ciphertext space (the length of the element is expressed as $|x|$), the public key is an element of $\mathbb{G}_1$, and the signature can be regarded as an element of the general ciphertext space. Therefore, the communication cost of our solution is $(n+5)|\mathbb{G}_1| + 5|\mathbb{Z}_p^*| + 1|x| + 2|ID| + 2k + 1l$.

Table 3 and Figure 5 show the comparison of communication cost. We assume that the size of the message sent in the alliance chain is $|x| = 160$ bits. When we encrypt the ciphertext with a key length of 80 bits, the size of $q$ is 1024 bits. Therefore, the size of the element in $\mathbb{G}_1$ is 1024 bits, and the size of the element in $\mathbb{Z}_p^*$ is 2048 bits. However, we can use standard compression techniques [26] to reduce the size of elements in $\mathbb{G}_1$ to 520 bits (65 bytes), and the size of elements in $\mathbb{Z}_p^*$ is 1024 bits (128 bytes). In addition, the length of $|ID|$ is 8 bits, which occupies one byte, and the length of both $k$ and $l$ is 512 bits. When only one user is considered, the communication cost of scheme [22] is $8|\mathbb{G}_1| + 4|\mathbb{Z}_p^*| + 1|x| + 1|ID| + 3k + 2l = 8 \times 65 + 4 \times 128 + 1 \times 20 + 1 \times 1 + 3 \times 64 + 2 \times 64 = 1373$ bytes, the communication cost of scheme [23] is $(n+3)|\mathbb{G}_1| + (n+2)|\mathbb{Z}_p^*| + 1|x| + 4|ID| + 2k + 2l = 4 \times 65 + 3 \times 128 + 1 \times 20 + 4 \times 1 + 2 \times 64 + 2 \times 64 = 924$ bytes, and the communication cost of ours is $(n+5)|\mathbb{G}_1| + 5|\mathbb{Z}_p^*| + 1|x| + 2|ID| + 2k + 1l = 6 \times 65 + 5 \times 128 + 1 \times 20 + 2 \times 1 + 2 \times 64 + 1 \times 64 = 1244$ bytes. The proposed scheme's total communication cost is decreased by 10% compared with scheme [22]. The communication cost of scheme [23] is lower than ours, but this is only for a single user. As the number of users continues to grow, the advantages of the proposed scheme will be revealed.

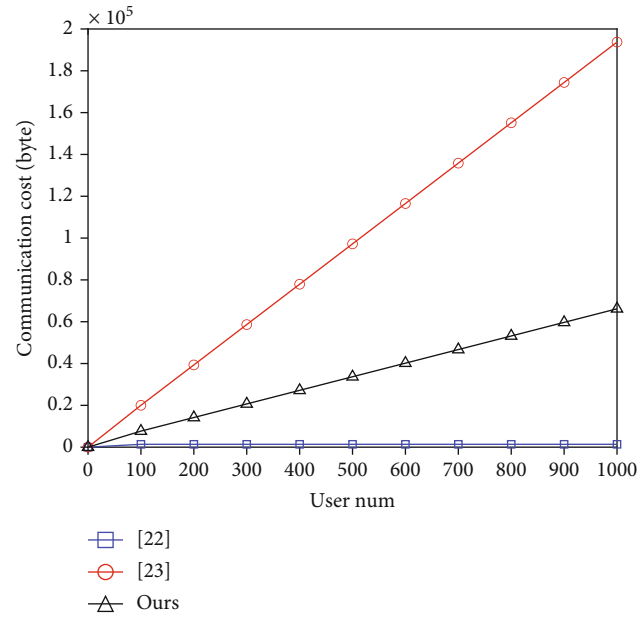


Figure 6: Impact of the number of users on calculation time.

With the increase of blockchain network users, the required calculation cost and communication cost will also increase. Next, we will analyze and compare the changing trend of the communication cost and calculation cost of our solution and other solutions when the user scale changes. The analysis results are shown in Figures 6 and 7. It can be seen from Figure 6 that the calculation costs of [22] and ours continue to increase with the number of users, but the growth rate is much lower than the solution proposed in [23]. It can be seen from Figure 7 that the communication cost of the solution proposed in [22] is not affected by the number of users $n$ and always remains at a low level. The communication costs of [23] and ours are linearly positively correlated with the

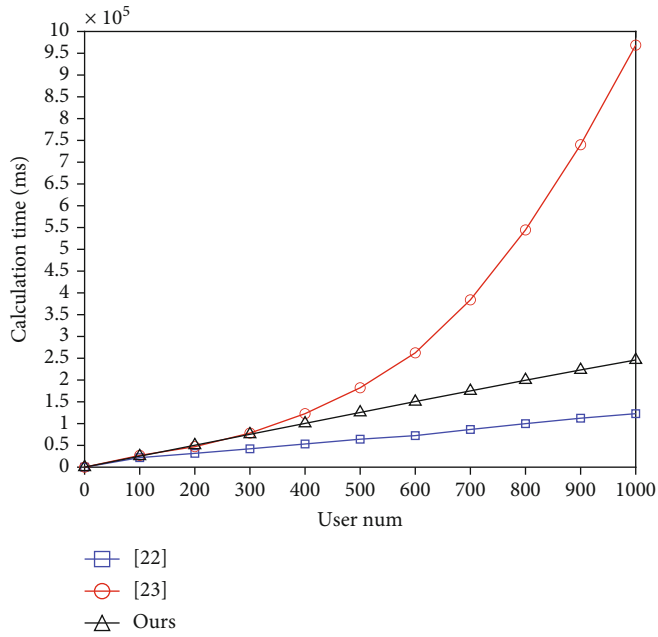


Figure 7: Impact of the number of users on communication cost.

number of users $n$. However, it can be seen that the communication cost of [23] is the highest, and the gap with the other two schemes is also getting bigger with the continuous increase of $n$. In conclusion, the proposed scheme has higher comprehensive performance.

## 6. Conclusion

The basic features of blockchain technology make it very suitable for data protection and sharing. This paper proposes a blockchain-based insurance claim data sharing model. For example, the insurance company can access the user's auto maintenance record through proxy reencryption technology and realize multiuser data sharing while protecting data privacy. The analysis results show that the proposed scheme meets many security requirements and has higher comprehensive performance compared with the existing two schemes.

## Data Availability

All data included in this study are available upon request by contact with the corresponding author.

## Conflicts of Interest

The authors declare that there are no conflicts of interest concerning the publication of this paper.

## Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities of Southwest Minzu University (No: 2020NYB17), the Fund of Guangxi Key Laboratory of Cryptography and Information Security (No: GCIS202121), the National Natural Science Foundation of China (No: 61976047), the Key Fund Project of Sichuan Provincial Department of Education (No: 17ZA0414), and the Sichuan Science and Technology Program (No: 2017JY0230).

## References

[1] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, "Blockchain technology: beyond bitcoin," *Applied Innovation Review*, vol. 2, pp. 1–19, 2016.

[2] L. J. Kish and E. J. Topol, "Unpatients–why patients should own their medical data," *Nature Biotechnology*, vol. 33, no. 9, pp. 921–924, 2015.

[3] H. Watanabe, S. Fujimura, A. Nakadaira, Y. Miyazaki, A. Akutsu, and J. K-ishigami, "Blockchain contract: securing a blockchain applied to smart contracts," in *2016 IEEE International Conference on Consumer Electronics*, pp. 467-468, Las Vegas, USA, 2016.

[4] L. Q. Zhao, "The analysis of application, key issues and the future development trend of blockchain technology in the insurance industry," *American Journal of Industrial and Business Management*, vol. 10, no. 02, pp. 305–314, 2020.

[5] D. Popovic, C. Avis, M. Byrne et al., "Understanding blockchain for insurance use cases: a practical guide for the insurance industry," *British Actuarial Journal*, vol. 25, no. 13, pp. 1–23, 2020.

[6] A. Bogner, M. Chanson, and A. Meeuw, "A decentralised sharing app running a smart contract on the ethereum blockchain," in *The 6th International Conference on the Internet of Things*, pp. 177-178, New York, USA, 2016.

[7] A. E. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: the blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE Symposium on Security and Privacy*, pp. 839–858, San Jose, USA, 2016.

[8] A. Ekblaw, A. Azaria, J. D. Halamka, and A. Lippman, "A case study for blockchain in health care: \med "rec" prototype for

electronic health records and medical research data," in *2016 International Conference on Open and Big Data*, pp. 1–13, Bethesda, USA, 2016.

[9] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018.

[10] A. Roehrs, C. A. da Costa, R. da Rosa Righi, V. F. da Silva, J. R. Goldim, and D. C. Schmidt, "Analyzing the performance of a blockchain-based personal health record implementation," *Journal of Biomedical Informatics*, vol. 92, pp. 103140–103140, 2019.

[11] J. Hua, G. Shi, H. Zhu, F. Wang, X. Liu, and H. Li, "CAMPS: efficient and privacy-preserving medical primary diagnosis over outsourced cloud," *Information Sciences*, vol. 527, pp. 560–575, 2020.

[12] D. Q. Fu and L. Fang, "Blockchain-based trusted computing in social network," in *The 2nd IEEE International Conference on Computer and Communications*, pp. 19–22, Chengdu, China, 2016.

[13] X. G. Liu, "A blockchain-based medical data sharing and protection scheme," *IEEE Access*, vol. 7, pp. 118943–118953, 2019.

[14] M. Mettler, "Blockchain technology in healthcare: the revolution starts here," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services*, pp. 1–3, Munich, Germany, 2016.

[15] Z. Zheng, S. A. Xie, H. N. Dai, X. P. Chen, and H. M. Wang, "An overview of blockchain technology: architecture, consensus, and future trends," in *2017 IEEE International Congress on Big Data*, pp. 557–564, Honolulu, USA, 2017.

[16] M. A. Khan and K. Salah, "IoT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[17] Y. Yuan and F. Y. Wang, "Blockchain: the state of the art and future trends," *Acta Automatica Sinica*, vol. 38, pp. 68–75, 2016.

[18] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 127–144, Zagreb, Croatia, 1998.

[19] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," in *2007 ACM Conference on Computer and Communications Security*, pp. 185–194, Alexandria, USA, 2007.

[20] L. F. Guo and B. Lu, "Efficient proxy re-encryption with keyword search scheme," *Journal of Computer Research and Development*, vol. 51, no. 6, pp. 1221–1228, 2014.

[21] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[22] Y. X. Ji, *Blockchain-based user location information security sharing scheme*, Master's thesis, Xidian University, 2019.

[23] S. Lanlan, *Research on cloud storage and sharing of re-encrypted ciphertext based on block chain attribute agent*, Master's thesis, Jiangxi University of Science and Technology, 2019.

[24] J. H. Zhang and J. Mao, "An efficient RSA-based certificateless signature scheme," *Journal of Systems and Software*, vol. 85, no. 3, pp. 638–642, 2012.

[25] J. W. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wireless body area networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 332–342, 2014.

[26] K. A. Shim, "CPAS: an efficient conditional privacy-preserving authentication scheme for vehicular sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 4, pp. 1874–1883, 2016.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6039900, 16 pages
https://doi.org/10.1155/2021/6039900

*Research Article*

# 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



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 overestimate 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 preciseness 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 (WMEWMA), 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]:

$$\mathrm{ETX} = \frac{1}{d_f \times d_r}. \tag{1}$$

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.

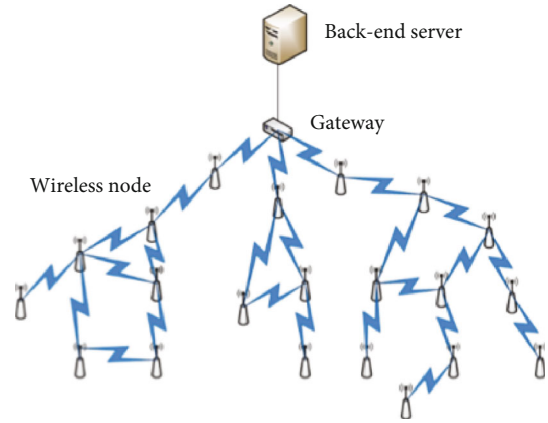


Figure 1: Network topology.

*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*. *P*1 is the delivery rate from *S* to *M*, *P*2 is the delivery rate from *M* to *D*, *P*3 is the delivery rate from *S* to *N*, and *P*4 is the delivery rate from *N* to *D*. If calculated using PRR, the final success rate of *R* 1 is 1/3 and *R*2 is 1/4, so choose *R*1. If ETX or DBETX is used, the sum of ETX in *R*1 is 4 and the sum of ETX in *R*2 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).

$$\prod_{i=1}^{n} p_i > \prod_{j=1}^{m} q_j, \tag{2}$$

$$\sum_{i=1}^{n} \frac{1}{p_i} < \sum_{j=1}^{m} \frac{1}{q_j}. \tag{3}$$

Formula (2) and Formula (3) in $p_i$ and $q_j$ are 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

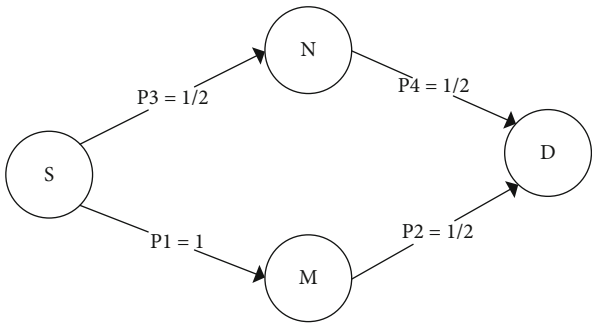


Figure 2: PRR Comparison with ETX on routing.

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.

$$\mathrm{PRR} = \begin{cases} 95\% & \mathrm{RSSI} < 65 - 0.000157769434713252 3\mathrm{LQI}^3 + 4.708425273599874\mathrm{LQI} - 214.9115458068067(\%) \\ 65 \le \mathrm{RSSI} \le 95, & \\ 5\% & \mathrm{RSSI} > 95, \end{cases} \tag{4}$$

$$\mathrm{LQI} = 5.3145 \times \mathrm{SNR} + 94.0477. \tag{5}$$

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

**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: $K \longleftarrow (P + R)$
4: $LQI \longleftarrow LQI + K * (X - LQI)$
5: $P \longleftarrow P - K * (P + R) * K$
6: **end while**
7: **return** $LQI$

Algorithm 1: One-dimensional Kalman filter algorithm.

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 |

the estimation process, each node sends the probe packet and also receives the probe packet from other nodes and records the relevant data. Because the gateway is the destination for all nodes to send data, it receives the probe packet from the neighbor node and records the information, without sending any probe packet itself. The sending of the probe packet requires a certain time interval and a certain number. The format of the probe packet is shown in Table 2. When a node receives a probe message, it can know how many probe messages it will send and the sequence number of the current probe message. In this way, the software-based link quality estimation is obtained. The hardware-based link quality estimation has been read from the underlying hardware when the packet is received. However, because the probe packet should be sent at a certain time interval, and the probe packet needs to reach a certain number, the process of sending the probe packet has taken up a certain time and produced a certain energy consumption. The results of this approach can lead to better link performance, such as higher delivery rates, lower latency, and lower power consumption.

### 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 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 nexthop 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.

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 |

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

Algorithm 2: Link quality information collection algorithm.

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 |

*3.5.2. Reliable Routing Algorithm.* The main function of 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

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

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

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

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

Algorithm 3: Generation algorithm of routing information packet.

**Require:** Total number of probe packets sum and received from each neighbor node count recorded
**Ensure:** The neighbor node receives its own packet rate
1: Initialize: Substitute *LQI* after Kalman filter into Formula (4) to get $PRR_{LQI}$;
2: *PRR*←*count*/*sum*
3: **if***sum*>*100***then**
4: $\alpha \longleftarrow 1$
5: **else if***sum* >*50***then**
6: $\alpha \longleftarrow sum/100$
7: **else**
$\alpha \longleftarrow 0.5$
8: **end if**
9: $PPR \longleftarrow \alpha * PPR + (1 - \alpha) * PPR_{LQI}$

Algorithm 4: Package reception rate estimation.

$$\frac{\partial P_{\text{delivery}}}{\partial n} = -(1-p)^n \times \ln(1-p) = (1-p)^n \ln \frac{1}{1-p}. \tag{8}$$

Because $p$ is less than or equal to 1, the results of both partial derivatives are nonnegative. Formula (7) shows that when the $n$ is constant, the derivative decreases with the increase of the $p$, which indicates that the lower the success rate, the greater the effect of the retransmission mechanism, whereas the higher the success rate, the smaller the effect of the retransmission mechanism. At the same time, it shows that the higher the success rate PRR, the smaller the impact of the same degree of error on the final result as link quality estimation. 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 $p$ 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 most reliable path from vertex $S$ to vertex $T$. The optimal path is the path $P$ from vertex $v_{i,j}$ to a certain victorious vertex $v_{i,k}$ in the decision graph $D = (V, E)$. If the cost is the least among all victorious vertices from $v_{i,j}$ to $v_{i,k}$, then $P$ is the optimal path of vertex $v_{i,j}$. According to the optimization principle, let $P = v_{i,j}v_{i+1,k}v_{i+2}, m \cdots v_{s-1,r}, v_{s,r}$ is an optimal path for vertex $v_{i,j}$, then $P = v_{i+1,k}v_{i+2,m} \cdots v_{s-1,r}v_{s,r}$ must be the optimal path for vertex $v_{i+1,k}$. 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

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

$$P_{\text{fail}}(n) = ((1-p)^{\max})^{n}. \tag{9}$$

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 $\text{node}_i$ in the network sends $\text{total}_i$ packets, and the number of packets successfully transmitted to the destination is $\text{delivery}_i$; then, the lost number is $\text{total}_i$ minus $\text{delivery}_i$. 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

$$r_{\text{delivery}}(i) = \frac{\text{delivery}_i}{\text{total}_i}, \tag{10}$$

$$r_{\text{loss}}(i) = 1 - \frac{\text{delivery}_i}{\text{total}_i}. \tag{11}$$

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: *optimal_success* ⟵ 0
2: *optimal_nexthop*←*nexthop*
3: *optimal_hop*←*mydepth*
4: **if***repair*=*true***and***source* ≠ *optimal_nexthop***or***source* ≠ *suboptimal_nexthop***then**
5: **return**
6: **end if**
7: **for***i*=0,1,2...*neighbor_num***do**
8: **if***address_info*[*i*]=*address***then**
9: *suc*←1−(1−*PRR*[*i*])$^{maxtrans}$ ∗ *success_ratio*
10: According to the *source* field of the routing packet, find the corresponding record in the routing result and update it. If not found, record the *source, suc, nexthop, pathlength* value of the routing packet
11: **if***optimal* ≠ *source***then**
12: *suboptimal_nexthop* ← *optimal nexthop*;
13: *suboptimal_success* ← *optimal success*;
14: *suboptimal_hop* ← *optimal_hop*
15: *suboptimal_threshold*←*optimal_threshold*
16: **end if**
17: *optimal_nexthoop*←*source*
18: *optimal_success*←*suc*
19: *optimal_hop*←*path_length*+1
20: $P_{fail}$ ⟵ $(1 - PPR[i])^{maxtrans}$
21: *optimal_threshold*⟵1
22: **while**$P_{fail} > 0.00001$**do**
23: $P_{fail}$ ⟵ $(1 - PPR[i])^{maxtrans}$
24: *optimal_threshold*⟵*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***p***do**
2: **if***p* ⟶ *address* = *optimal_nexthop***then**
3: *p* ⟶ *alive* ⟵ *false*
4: *n_number* ⟵ *n_number* − 1
5: **end if**
6: *p* ⟵ *p* ⟶ *next*
7: **end while**
8: **if** the suboptimal solution exists and is not the solution of the minimum hop route **then**
9: *optimal_nexthop* ⟵ *suboptimal_nexthop*
10: *optimal_success* ⟵ *suboptimal_success*
11: *optimal_hop* ⟵ *suboptimal_hop*
12: *optimal_threshold* ⟵ *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: *repair* ⟵ *true*
18: sent routing packets to neighboring nodes
19: **return***optimal_nexthop*

Algorithm 6: Node failure recovery mechanism.

Table 5: Hardware environment and simulation environment.

| | |
|---|---|
| CPU | I5-4590 3.3 GHz |
| Memory | 4 GB 1600 MHz |
| Network simulator | OMNeT++5.0 |
| OS | Windows 8.1 |

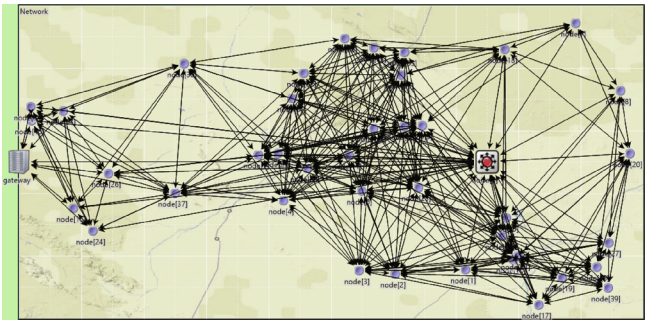


Figure 3: Simulation topology.

and packet loss rate $r_{\text{loss}}$ is

$$r_{\text{delivery}} = \frac{\sum_{i=1}^{n} \text{delivery}_i}{\sum_{i}^{n} \text{total}_i}, \tag{12}$$

$$r_{\text{loss}} = 1 - \frac{\sum_{i=1}^{n} \text{delivery}_i}{\sum_{i}^{n} \text{total}_i}. \tag{13}$$

(2) Network average end-to-end delay

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

$$\text{delay}(i) = \frac{\sum_{j-1}^{\text{total}_i} t_{D,i}(j) - t_{S,i}(j)}{\text{total}_i}, \tag{14}$$

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

(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.

Table 6: Specific parameters of the simulation topology.

| Parameter name | Value |
|---|---|
| Industrial site length | 100 m |
| Industrial site width | 50 m |
| Number of site equipment | 40 |
| Limit of communication distance | 30 m |

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

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:

$$\text{reliability} = \frac{\sum_{i=1}^{n} R(i)}{n},$$
$$R(i) = \begin{cases} 1 r_{\text{delivery}}(i) \geq 95\%, \\ 0 r_{\text{delivery}}(i) < 5\%. \end{cases} \tag{17}$$

*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.

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 |

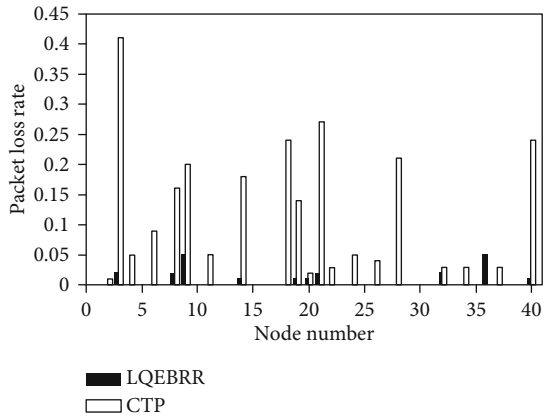


Figure 4: Packet loss ratio of every node.

*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.

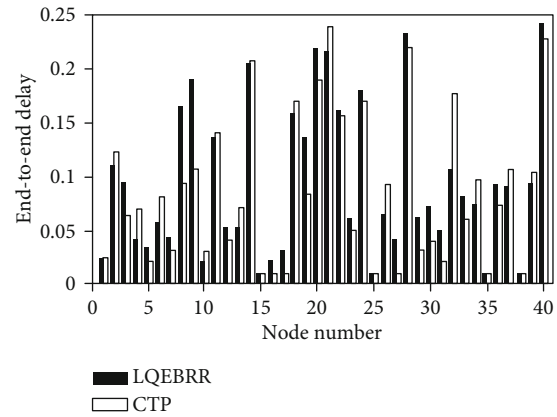


Figure 5: End-to-end delay of every node.

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

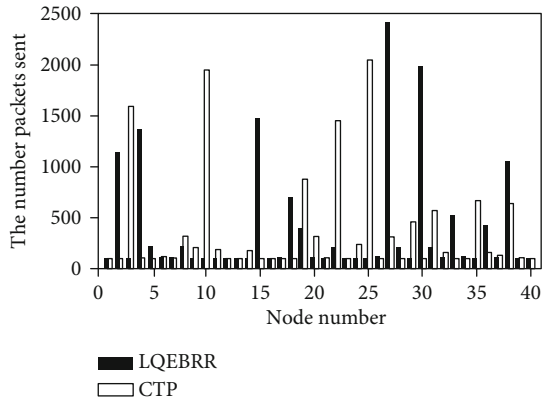


Figure 6: Transmission times of every node.

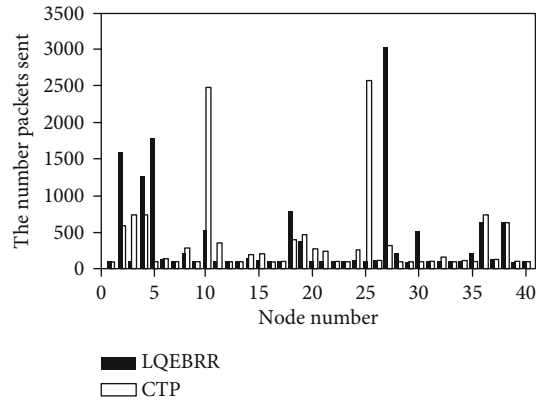


Figure 9: Transmission times of every nodes.

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 |

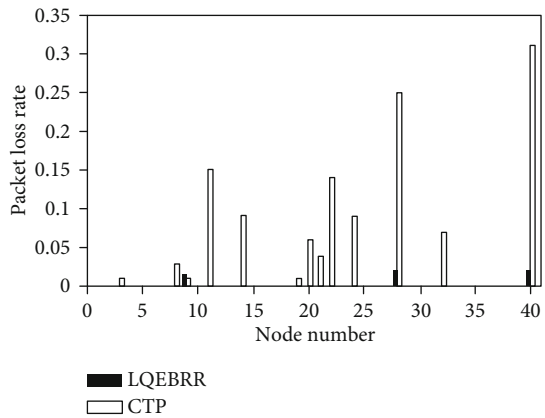


Figure 7: Packet loss ratio of every node.

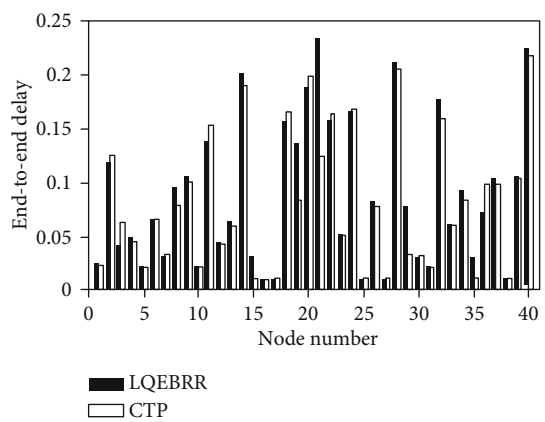


Figure 8: End-to-end delay of every nodes.

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

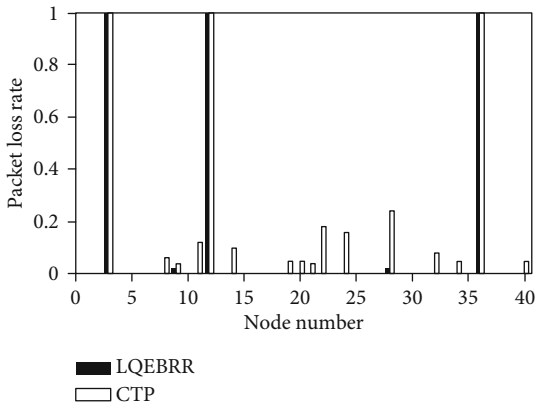


Figure 10: Packet loss ratio with failure at begin.

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.

Table 10: Performance of the entire network with failure at begin.

| Routing mechanisms | Average delivery rate | Average end-to-end delay | Number of total packets |
|---|---|---|---|
| HLQEBRR | 92.3% | 0.0819 s | 14969 |
| CTP | 89.6% | 0.0808 s | 13313 |

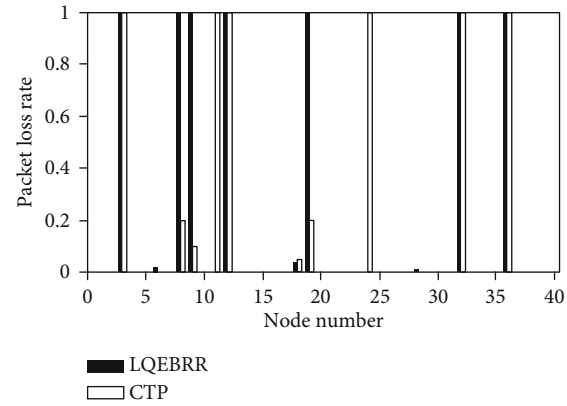


Figure 11: Failure in midway without failure recovery.

Table 11: Performance of the entire network with failure in midway but no restore.

| Routing mechanisms | Average delivery rate | Average end-to-end delay | Number of total packets |
|---|---|---|---|
| HLQEBRR | 82.4% | 0.0629 s | 13039 |
| CTP | 83.7% | 0.0624 s | 12980 |

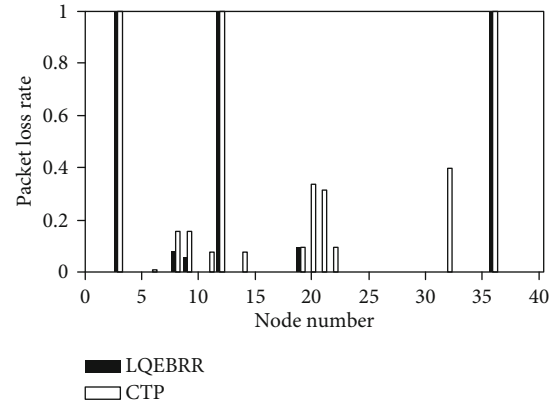


Figure 12: Failure in midway with failure recovery.

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 CTP 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 (2020MS086), 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 wirelesshart 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. Tuset-Peiro, 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-202135th 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.

[25] X. Gong, *Graph Theory and Network Optimization Algorithm*, Chongqing University Press, 2009.

[26] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Acm Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2009.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 6623052, 11 pages
https://doi.org/10.1155/2021/6623052

*Research Article*

# Predicting Customer Turnover Using Recursive Neural Networks

**Abdullah Jafari Chashmi**,[1] **Vahid Rahmati**,[2] **Behrouz Rezasoroush**,[3] **Masumeh Motevalli Alamoti**,[4] **Mohsen Askari**,[3] **and Faezeh Heydari Khalili**[5]

[1]*Department of Electrical Engineering, Mahdishahr Branch, Islamic Azad University, Mahdishahr, Iran*
[2]*Faculty of Information Technology, Faculty of Computer, Payame Noor Assaluyeh, Bushehr, Iran*
[3]*Department of Electrical and Computer Engineering, Faculty of Molasadra Branch, Technical and Vocational University (TVU), Ramsar, Iran*
[4]*Department of Computer Engineering, Karaj Beranch, Islamic Azad University, Karaj, Iran*
[5]*Department of Computer Engineering, Ramsar Beranch, Islamic Azad University, Ramsar, Iran*

Correspondence should be addressed to Mohsen Askari; m.askari@iauramsar.ac.ir

Received 27 November 2020; Accepted 17 May 2021; Published 7 June 2021

Academic Editor: Jianhui Lv



The most valuable asset for a company is its customers' base. As a result, customer relationship management (CRM) is an important task that drives companies. By identifying and understanding the valuable customer segments, appropriate marketing strategies can be used to enhance customer satisfaction and maintain loyalty, as well as increase company retention. Predicting customer turnover is an important tool for companies to stay competitive in a fast-growing market. In this paper, we use the recurrent nerve sketch to predict rejection based on the time series of the lifetime of the customer. In anticipation, a key aspect of identifying key triggers is to turn off. To overcome the weakness of recurrent neural networks, the research model of the combination of LRFMP with the neural network has been used. In this paper, it was found that clustering by LRFMP can be used to perform a more comprehensive analysis of customers' turnover. In this solution, LRFMP is used to execute customer segregation. The objective is to provide a new framework for LRFMP for macrodata and macrodata analysis in order to increase the problem of business problem solving and customer depreciation. The results of the research show that the neural networks are capable of predicting the LRFMP precursors of the customers in an effective way. This model can be used in advocacy systems for advertising and loyalty programs management. In the previous research, the LRFM and RFM algorithms along with the neural network and the machine learning algorithm, etc., have been used, and in the proposed solution, the use of the LRFMP algorithm increases the accuracy of the desired.

## 1. Introduction

Nowadays, not all customers have the same importance for companies, and companies are looking to identify and analyze customer attributes, as well as segregation and clustering based on their value. Detecting, analyzing the characteristics, and clustering of customers based on the value they have for the company provide the context for the optimal allocation of limited resources, the use of appropriate marketing strategies, and, ultimately, profitability management along with customer relationship management. The value of the life cycle of a customer is a concept that can help companies in this regard, which is determined by using different models. Customer analysis is a process that uses customer behavior data to help key business decisions through market segmentation and forecasting analysis.

Customer analysis is a process that uses customer behavior data to help key business decisions through market segmentation and forecasting analysis. This information is used by companies for direct marketing, site selection, and customer relationship management. Customer analysis plays an important role in predicting customer behavior. Life-style models use different strategies for modeling customer behavior. One of the most prominent uses of variable, sequencing,

and monetary is (RFM) variables [1–3]. These variables provide a relative understanding of customer behavior and try to answer these questions:

(i) When did the customer buy the last time?

(ii) How often do they buy?

(iii) How much do they spend?

RFM variables are good statistics for customer behavior modeling and the main argument in the industry because it can be easily implemented [1, 4].

By blasting large data and accessing online and offline transaction data, modeling the true value of customer life and predicting customer behavior by using RFM factors leads to corporate earnings, market profits, and greater customer loyalty [1, 5].

Life-style models use different strategies for modeling customer behavior.

Recent efforts have been made to predict customer behavior. To predict the restaurant's priority, an artificial neural network (ANN) model is proposed in [2].This model models the social media position control, customer historical priorities, the impact of the customer's social network, and the client's customer transfer characteristics. FM variables are used to find retailers' parts of the data from the transfer of electronic funds transfer at the point of sale (EFTPOS). Other researchers (9) extracted the customer buying behavior by finding relationships among products and taking advantage of customer motivation. Then, customer preferences for student attributes are identified through probabilistic models for matching products with customers. Artificial neural networks are also used to predict RFM of blood donors. In this strategy, a base version of artificial neural networks is used that considers time as a separate input variable [3, 6].

Disconnection prediction is a process in which customers identify who are turning away. In order to execute a turnaround forecast, a declined customer must be defined [4].

## 2. Review of Literature

In a study, Moslehi et al. focused on using the LRFM model to segment customer behaviors based on their life cycle value. Based on the CRISP data mining method and using the group hierarchical process, according to diagnostic analysis, customers are divided into 16 groups and 5 main clusters (under the headings of loyal customers, potential loyal customers, new, lost customers, and high-consumption customers). In this study group, 13 customers (loyal customers) with the highest value in terms of the life of the company's purchase period were identified that the company should strive to maintain them [1].

Akhundzadeh Noghabi and his colleagues in a study identified customer behavioral groups and the characteristics of each of these groups in the telecommunications industry. For this purpose, they first grouped customers based on RFM variables and $K$-means method, then using association rules to identify customer behavior patterns in each group. Based on the results of this study, seven different behavioral groups of customers were identified that these results can lead to a better view and understanding of customer behavior patterns and improve marketing strategies [5].

Baradaran and Biglari used the improved RFM model to segment customers in the manufacturing and distribution industries of popular goods. They improved the quality of segmentation by replacing the purchase sequence variable ($C$), which represents the customer's purchase sequence during a particular period and equal to the number of months of the year that the customer purchased during that period, with the purchase delay variable in the RFM model. The results showed that customer segmentation based on CFM is more accurate compared to the RFM model [5].

In a study, Khodabandeh Lou and Niknafs presented a new method for segmenting the customers of a grocery store based on their loyalty and defined appropriate strategies for each segment. In this study, the effect of several effective factors (including the number of purchased goods, number of returned goods, discount, and delay in distribution along with RFM variables) on increasing the quality of loyalty evaluation has been measured. Based on the results of this study, the researchers divided customers into five clusters in terms of loyalty (including loyal customers, potential loyal, new, lost, and turned away customers), and finally, appropriate strategies for managing each customer. The results showed that the developed RFM is very accurate in predicting customer loyalty [1].

Chang and Tsay in their research entitled "Combining SOM and K-means in cluster data mining" have proposed the LRFM model to mean increasing the duration of customer relationship, which after extracting the model data and clustering from a combination of two value matrices (combination of two FM indicators) and loyalty matrix (combination of two LR indicators) were used for analysis and classified customers into five types and sixteen categories. These researchers showed that adding this index improves the accuracy of identifying loyal customers [2].

Chang and Chen proposed a model based on a combination of RFM and $K$-means methods with hard set theory. Based on their model, they classified customer loyalty by determining the number of clusters (number of clusters 3, 5, and 7, respectively), then evaluated and described the characteristics of customers in each cluster, and evaluated and implemented CRM. [3].

By using the RFM model and $K$-means clustering method, Wu and Chang and Lu analyzed the value of customers of one of the industrial equipment manufacturing companies. After preparing the data, the customers were divided into six clusters based on RFM indices. Then, the characteristics of all customers in the form of clusters were analyzed using the evaluation of the customer's lifetime value. At the end of the research, appropriate suggestions have been made with different groups of customers [6].

In another study by Lee et al. using a two-stage clustering method to analyze customer characteristics in customer category management based on the LRFM model indices and using a two-stage clustering method (from the method of determining the optimal number of clusters and K-means

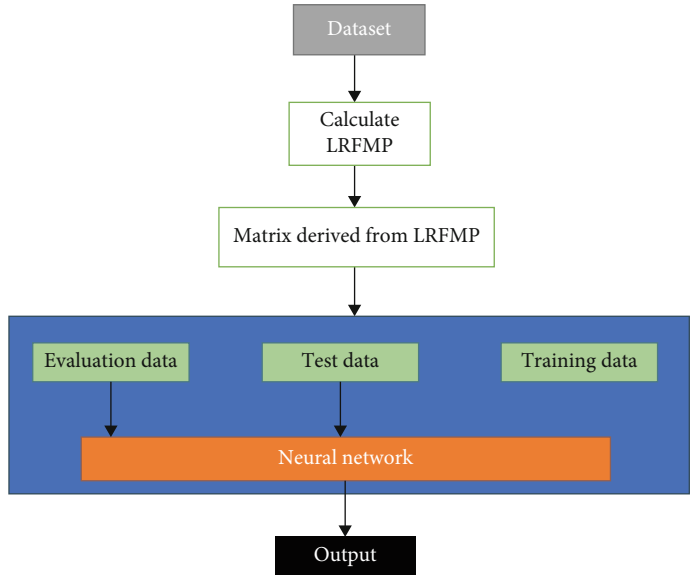


Figure 1: View of the proposed model.

method) analyzed customer characteristics to improve customer relationship management in the textile industry, the results of which created a better understanding in the company to determine marketing strategies. Also, in the industry surveyed in Taiwan, it was found that some customers have communication lengths that are longer and more loyal, but the volume of transactions and the frequency of those customers may not be high [4].

Wei and Lin and Weng focused on the application of the LRFM model in the well-being of dental clinic clients and finally identified loyal clients based on the LRFM model. According to the results, patients were divided into four groups (including loyal customers, active, new, and unknown), and appropriate strategies for dealing with each group of these customers were determined [7].

Salehinejad et al. [8] analyzed the function of the recurrent neural network with the ReLU activation function model (ReLU-RNN) along with LSTM-RNN and SRNN. Performance results show that recursive neural network models have competitive performance for the RFM recommender system. The ReLU activation function has almost better performance (approximately 80% accuracy) compared to LSTM and SRNN. ReLU-RNN performs better for latency (78%) and sequencing (82%). This model has a performance of 79% for the monetary value parameter, while the LSTM-RNN has a performance (80% in terms of the monetary value parameter); so, it has a better performance.

In another study by Salehinejad et al. [9], they systematically examined major recent developments in neural networks in the literature and introduced challenging problems in RNN training. RNN refers to an artificial neural network that has frequent connections between them. Frequent connections learn dependencies between consecutive data or input time series. The ability to learn sequential dependencies allows RNNs to gain popularity in applications such as speech recognition, speech synthesis, machine vision, and video description generation. One of the main challenges of training RNNs is learning long-term data dependencies. This is generally due to the large number of parameters that must be optimized during RNN training over long periods [7]. In this study, they discussed the RNN architecture and various training methods for it that can be used to solve problems related to RNN training.

## 3. LRFMP Model

The LRFM model is a method used for customer clustering in customer relationship management (CRM). In this model, customers are categorized based on four characteristics of customer relationship length, purchasing novelty, buying frequency, and purchasing value. Based on this model, all four features of the LRFM term are derived from ($R$), sequence ($F$), and polynomial ($M$) [8].

Classic LRFM models have mostly performed well in customer segmentation in many different industries [9–11]. This study contributes to prior literature by proposing a new RFM model, called LRFMP for the customer segmentation and providing useful insights about behaviors of predicting costumer turnover. Clustering by LRFMP can be used to perform a more comprehensive analysis of roaming clients. This algorithm provides meaningful and valid categories and predicts that there are patterns for data set separation.

According to Rinartz and Kumar [12], Chang and Tesa [13], and Lee et al. [10], the RFM model cannot provide customers with long-term relationships and customers with short-term relationships with the organization that identifies in their research, and they propose the idea of customer relationship length and explore its impact on customer loyalty and profitability. They say that increasing customer relationship length will improve customer loyalty. It defines the variable that represents the time interval between the first and last customer purchases in the observed interval. The RFM model offers customers who have recently created high financial value for the company and have short-term purchasing

Table 1: Part of the data used.

| Customer ID | Age | Address | Product subset | Product ID | Value | Property | Sales Price | Transaction date |
|---|---|---|---|---|---|---|---|---|
| 00141833 | F | F | 130207 | 47101050 | 2 | 44 | 52 | 2001/01/01 |
| 01376753 | E | E | 110217 | 47102657 | 1 | 150 | 129 | 2001/01/01 |
| 01603071 | E | G | 100201 | 47120191 | 1 | 35 | 39 | 2001/01/01 |
| 01738667 | E | F | 530105 | 47101687 | 1 | 94 | 119 | 2001/01/01 |
| 02141497 | A | B | 320407 | 47104313 | 1 | 100 | 159 | 2001/01/01 |
| 01868685 | J | E | 110109 | 47100435 | 1 | 144 | 190 | 2001/01/01 |

patterns over the average purchase frequency among repeat-bought customers as valued customers, while the factor of the length of communication with the company is ignored. The length of customer relationship with the organization reflects the length of time a customer has started communicating with the organization. The article states that the length of customer relationship with the organization has a positive relationship with the probability of its future relationship. The LRFM clustering model has often worked well in customer segmentation in various industries, but we have included in this article the customer visit period ($P$) to the LRFM core model to determine customer behavior and their degree to measure. The periodicity of visiting customers is defined by ($P$), retirement time with ($R$), length ($L$), frequency ($F$), and monetary characteristics ($M$) [14, 15].

The length of the time interval per day is between the first and last customer visits. This shows customer loyalty, and the longer it is, the more loyal the customer is. The retirement time shows the waiting time, how it updates the customer engagement with the company, and repeats information about the buying trend [4, 16]. The traditional LRFM model is usually calculated as the time interval (usually per day) between the date of the last visitor's visit and the last date of the observation period. The variable in our model varies as the average number of days between the recent hits of the customer and the last date of the observation period [17, 18]. Thus, the experimental value in our model is calculated using the following equation:

$$\text{Recency}(n) = \frac{1}{n}\sum_{i=1}^{n} \text{date\_diff}(t_{\text{enddate}}, t_{m-I+1}). \quad (1)$$

In relation (1), $t_m - i + 1$, tm shows the last customer visit.

$n$ is the number of recent visitors to the client.

$t_m$ is the last visit from the customer between the dates and the end of the observation period, the extension, and the date of the visit to the customer near to the renewal.

Note that for $n = 1$, this newly predicted predictive variable is transformed into traditional experience, and therefore, the recency property covers our change.

Frequency refers to the total number of customer visits during the observation period. The higher the frequency becomes, the more customer loyalty. The monetary factor refers to the average amount of money the customer pays during visiting during the visit and represents the customer's share in the income of a company. A higher monetary value represents a larger share of the company [17, 19].

Table 2: Sample LRFMP output.

| Customer | $L$ | $R$ | $F$ | $M$ | $P$ |
|---|---|---|---|---|---|
| 1069 | 328 | 5815.455 | 11 | 2660 | 131.2672 |
| 1113 | 331 | 3553.889 | 18 | 3209 | 110.2734 |
| 1250 | 302 | 4569.286 | 14 | 1786 | 103.6006 |
| 1359 | 333 | 21323.33 | 3 | 2169 | 120.9605 |

## 4. Proposed Model

The proposed model will be based on prediction of time series data with a combination of clustering and neural network. In this model, first, the data in the preprocessing process are clustered using the LRFM base model as LRFMP, then the resulting data set is applied to the neural network based on the testing scenarios, and the results are reviewed and displayed. Figure 1 shows the proposed model.

We explain the proposed model in the following steps.

The first step includes the following actions.

The first step is to first clear the data and unnecessary details at this stage. Some customer records are discarded because they are not in accordance with the proposed method. In order to find the required data, the factors of length ($L$), latency ($R$), sequence ($F$), monetary ($M$), and periodic ($P$) are calculated from the total data. In our article, we have used the LRFMP solution, which has only $L$, $R$, $F$, $M$, and $P$ metrics.

These factors are considered for all customers of the initial aggregates, and all of the customers, whose total counts 817,741, are applied and those that have a lower value than the minimum specified value of each criterion. The customer segment is considered and displayed as an output that the number of customers dropped at this stage. At this stage (LRFMP model), there are 32268, which will be the number of inputs of the neural network in the next step.

Next, in this model, we have applied data clustering based on the LRFMP model. The LRFMP model is derived from the LRFM base model, with the $P$ attribute added, and using the purchase history, 9 continuous LRFMP features for each member will be calculated. Additionally, existing days from the registration date for each member are removed from the demographic information.

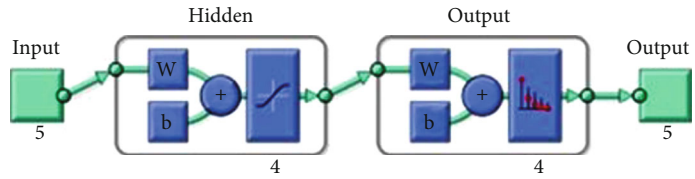


Figure 2: Proposed model with 4 hidden layers.

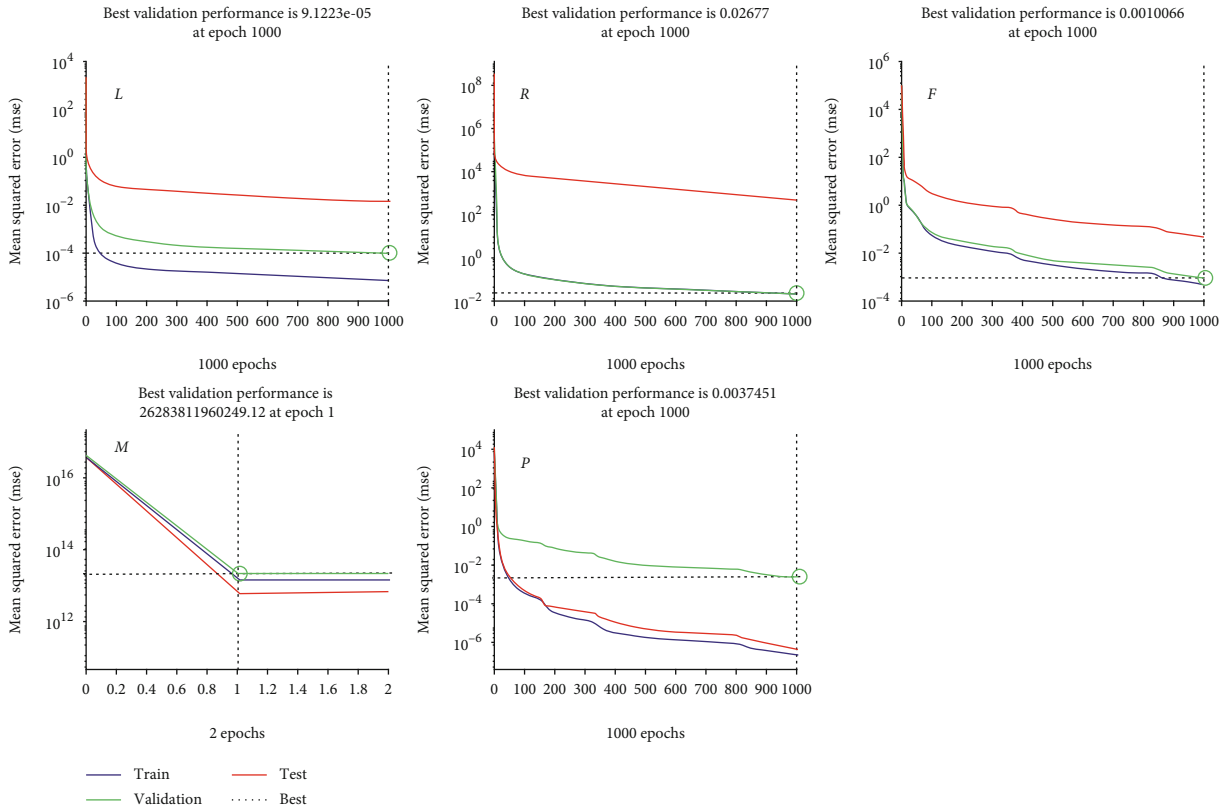


Figure 3: The graph of the neural network performance of model one.

$$\text{Periodicity} = \text{stdev}(\text{IVT}_1, \text{IVT}_2, \cdots, \text{IVT}_{n-1}, \text{IVT}_n). \quad (2)$$

The new feature shown in the relationship (2) shows whether customers regularly visit the stores.

Next, we set a period as the standard deviation of customer visit time (relationship (3))

$$\text{IVT}_i = \text{date}_{\text{diff}(t_{i+1}, t_i)}. \quad (3)$$

In Equations (2) and (3), IVT shows the time between visits by clients.

$n$ represents the number of clients the values between the visit time.

IVT is the time elapsed between the two consecutive customer reviews.

Thus, it is defined that where $i \geq 1$ and $t_i$ represent the date of visit $i$ from the customer. This quarterly benchmark shows that customer watches occur at regular intervals. If a customer has a low amount of dispersion, this means that the client is relatively seasonal or purchasing and can be identified on a regular basis.

In the second stage, the data obtained from the first stage are applied to the neural network and are based on different state of the results. At this stage, the neural network is defined based on a linear progressive network, and the results are obtained based on the number of neurons in the hidden layer. The proposed model is based on different scenarios. At this stage, the LRFMP model is first implemented, and then the resulting data are applied to the neural network based on the following structure.

(i) 50% of the resulting data as training

(ii) 25% of the resulting data as a test

(iii) 25% of the resulting data as an assessment

*4.1. Create a Model Using Neural Networks.* The neural network is a very powerful tool for classifying and predicting that is part of modern learning methods. Neural networks

Table 3: Model one neural network performance output.

| | $L$ | $R$ | $F$ | $M$ | $P$ |
|---|---|---|---|---|---|
| 4-layerneural network | 9.1223$e$-05 | 0.02677 | 0.0010066 | 26283811960249.12 | 0.0037451 |

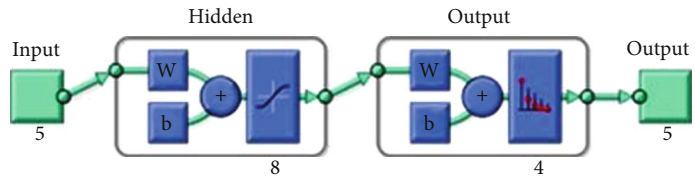


Figure 4: 8-layer model.

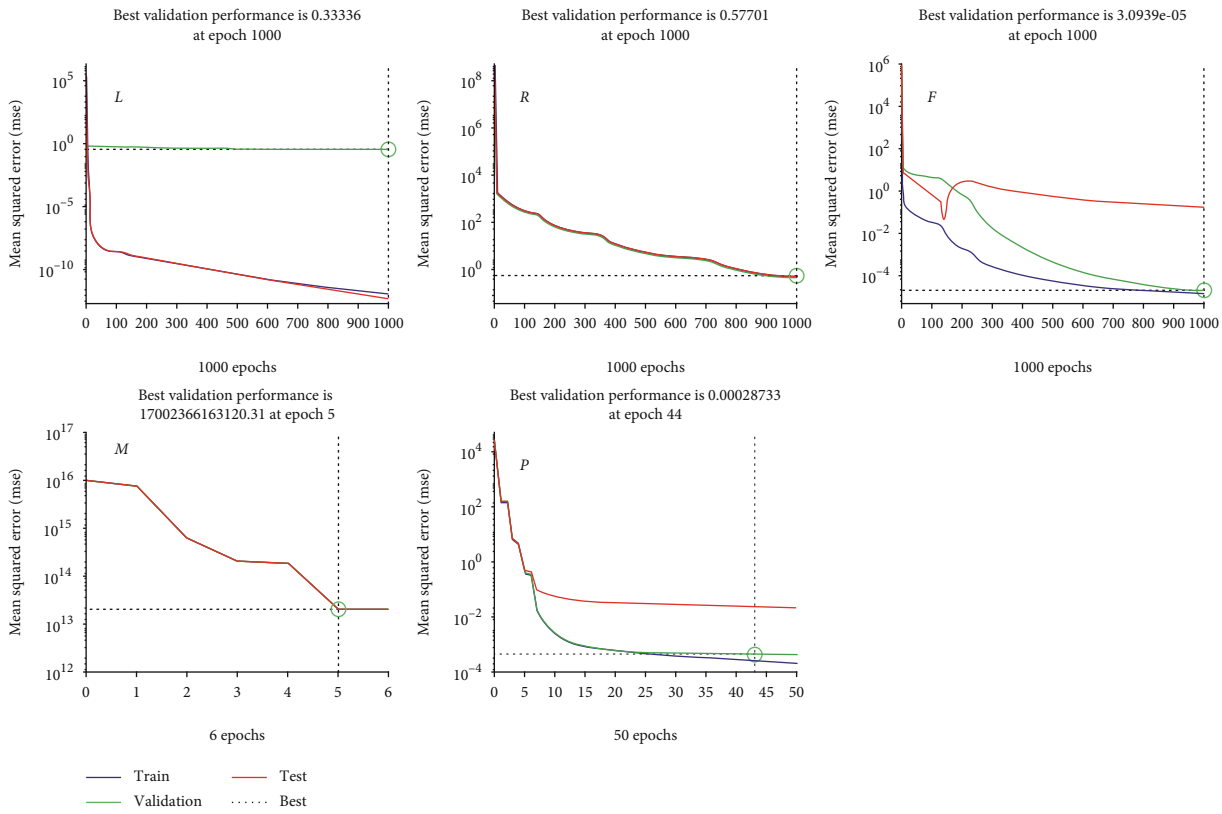


Figure 5: The graph of the neural network performance of model two.

are able to identify relationships between existing data and display those relationships. Neural network inputs are variables, and the outputs are states that we need to predict or control. Neural networks play two main functions, which are learning and calling. Learning refers to the weighting step of links in a neural network that enables them to generate an output vector in response to when the vector is stimulated by the input layer.

Calling is the step of accepting an input, excitability, and generating a response as output in order to generate a network structure. The system is made up of a large number of highly interconnected processing elements called neural networks that work together to solve a problem and transmit information through synapses (electromagnetic communications).

## 5. Simulation of the Proposed Method

The data used in this study was collected from the recsyswiki database. This database in the ta-feng section contains information about retail market data from an anonymous Belgian vendor, and we will selectively analyze this data in our studies.

Before starting the analysis, the collected data should be prepared by applying preprocessing methods. For this purpose, in the first stage, by examining the collected data, it is observed that some of them are not suitable as input in applying the proposed model. Therefore, it is necessary to use factors to limit the data contained in the data set. Some of the factors considered in this research are age, place of residence,

Table 4: Output related to neural network efficiency of the second model.

| | $L$ | $R$ | $F$ | $M$ | $P$ |
|---|---|---|---|---|---|
| 8-layerneural network | 0.33336 | 0.57701 | 3.0939$e$-05 | 170023661631120.31 | 0.00028733 |

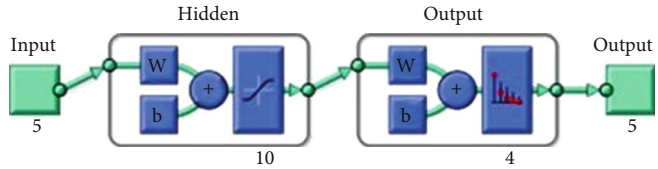


Figure 6: 10-layer model.

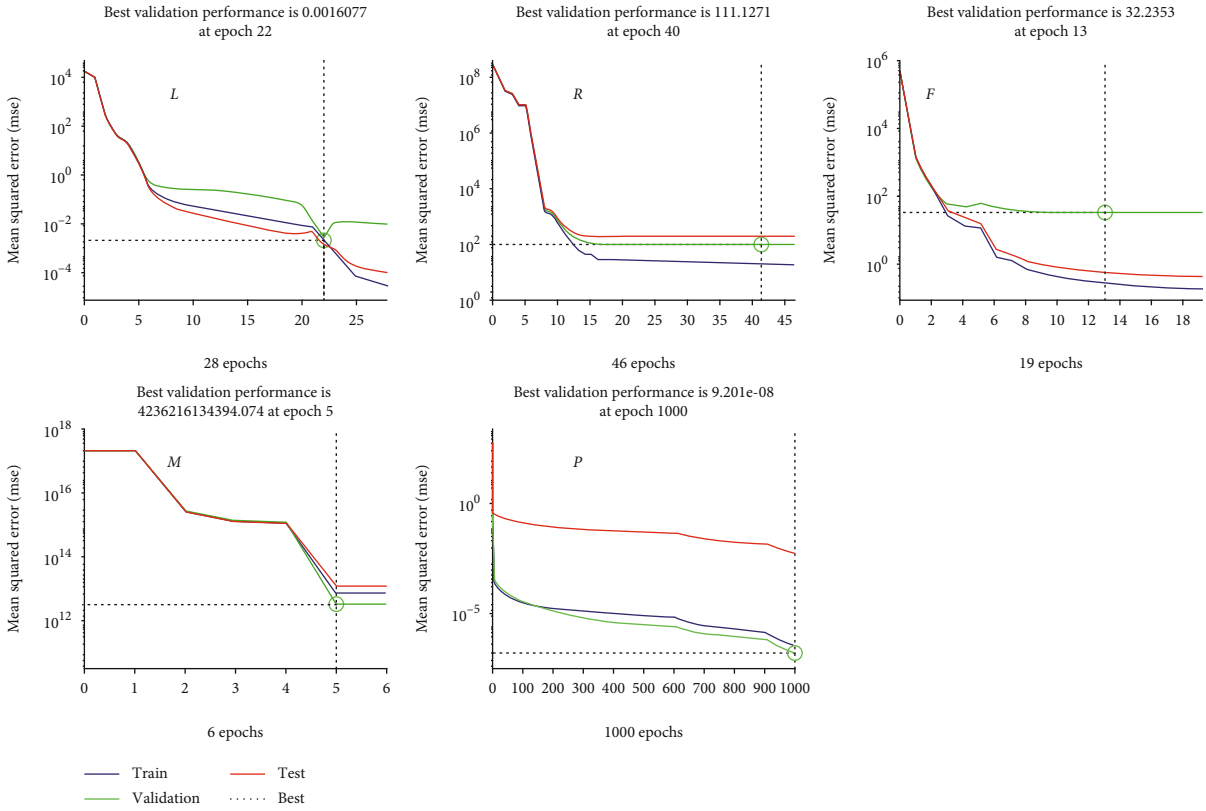


Figure 7: The graph of the neural network performance of model there.

defined subset for each product, product ID, and so on. In the proposed model, the primary data in the data set is limited by the LRFMP model, and finally, by applying sampling methods, we select the training and test sets. Feng is a food procurement suite published by ACM RecSys that includes products such as food and office supplies that relate to transactions made by users for 4 months (from November 2000 to February 2001) The total number of transactions in this collection is 817741, which includes 32266 users and 23812 products.

In Table 1, each record shows the products purchased by a customer each time they visit the store, which includes the fields of date and time of transaction, customer ID, age (in this study, we have 10 age categories), location, area zip code, the subsets defined for the product in question (which products in the store are categorized into different collections) and the unique identifier assigned to each collection, the amount that represents the number of subsets of the product purchased for a product by the customer is the amount of inventory that represents the remaining quantities of the product sold, and the selling price, which represents the price set as the consumer price to supply the product to the customer.

In the simulation performed in this research, a file called train.txt is considered as a training data set, which is a training set for the neural network. The training file contains all the data related to the transactions performed by the users in D, except for the last transaction performed by each user. user_tran is a file created from the train.txt data set that records each record in this file of additional information about users' next transactions.

Table 5: Model there neural network performance output.

| | $L$ | $R$ | $F$ | $M$ | $P$ |
|---|---|---|---|---|---|
| 10-layer neural network | 0.0016077 | 111.1271 | 32.2353 | 4236216134394.074 | 9.201$e$-08 |

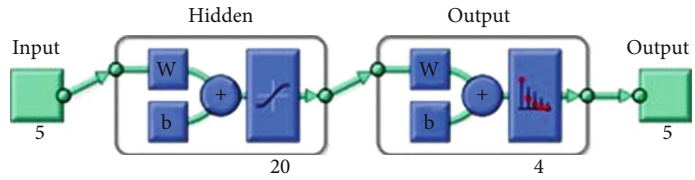


Figure 8: 20-layer model.

To identify deviant customers, in this study, we will follow the steps to discover the pattern:

(i) Collect real data related to customer characteristics and transactions made by them and store this data in the appropriate database

(ii) Reduce the volume of data collected by applying preprocessing methods and selecting optimal features. As a result of these measures, the dimensions and number of data properties are reduced

(iii) Apply different classifications or clustering algorithms to discover the pattern. Use of classification algorithms (to identify deviant customers)

The neural network is implemented based on 4 models of 4 layers, 8 layers, 10 layers, and 20 separate layers, the LRFMP output set is applied separately to the neural network, and the results are presented.

Table 2 shows an example of the output of LRFMP clustering calculations. As can be seen, for each client with a unique identifier, the length ($L$), endpoint ($R$), sequence ($F$), monetary ($M$), and periodic ($P$) characteristics are calculated and stored as a matrix.

The results of various models of the neural network are described below.

*5.1. The First Model.* Figure 2 displays the proposed model with 4 hidden layers used.

Figure 3 and Table 3 have shown the performance of the neural network of model 1 as outputs of length ($L$), backward ($R$), sequence ($F$), monetary ($M$), and periodic ($P$) outputs separately for training, testing, and evaluation data.

*5.2. Model Second.* In this model, 8 hidden layers are used, and Figure 4 shows the neural network model.

Figure 5 shows the performance of the neural network model 2 as outputs of length ($L$), backward ($R$), sequence ($F$), monetary ($M$), and periodic ($P$) output separately for training, testing, and evaluation data.

According to the diagrams in Figure 5, Table 4 shows the outputs related to the performance of the model 2 neural network.

*5.3. Model There.* This model uses 10 hidden layers, and Figure 6 illustrates this neural network model.

Figure 7 Function of the neural network of model 3 as outputs of length ($L$), back ($R$), sequence ($F$), monetary ($M$), and periodic ($P$) outputs separately for training, testing, and evaluation data.

The graphs of Figure 7 and Table 5 show the output of the neural network function of model 3.

*5.4. Model Four.* In this model, 20 hidden layers are used, and Figure 8 shows this model of the neural network.

Figure 9 shows the performance of the model 4 neural network as outputs of length ($L$), endpoint ($R$), sequence ($F$), monetary ($M$), and periodic ($P$) output separately for training, testing, and evaluation data.

According to the diagrams in Figure 9 and Table 6, the output of the neural network model 4 can be deserved.

## 6. The Performance of the Proposed Method

To compare the accuracy of the proposed algorithm, we use the test set data. The purpose of calculating accuracy is to measure the quality of the results obtained by applying the proposed algorithm and method in comparison with the actual results. Equation (4) is used to calculate the accuracy of the algorithms used.

$$\mathrm{Ac} = \left(\frac{N_{\mathrm{TP}} + N_{\mathrm{TN}}}{N_{\mathrm{TP}} + N_{\mathrm{TN}} + N_{\mathrm{FP}} + N_{\mathrm{FN}}}\right). \tag{4}$$

In relation (4), TP represents the number of samples that have been correctly identified as positive, in other words, the correct number of people who have been correctly identified as customers and who have been correctly identified as customers.

TN shows the number of samples that have been correctly identified as negative, in other words, the number of people who have not been identified as a reversing customer and in practice that have not been a reversed customer.

FP represents the number of samples that have been misdiagnosed as positive, in other words, the number of people who have not been turned away but have been identified as turning away customers.

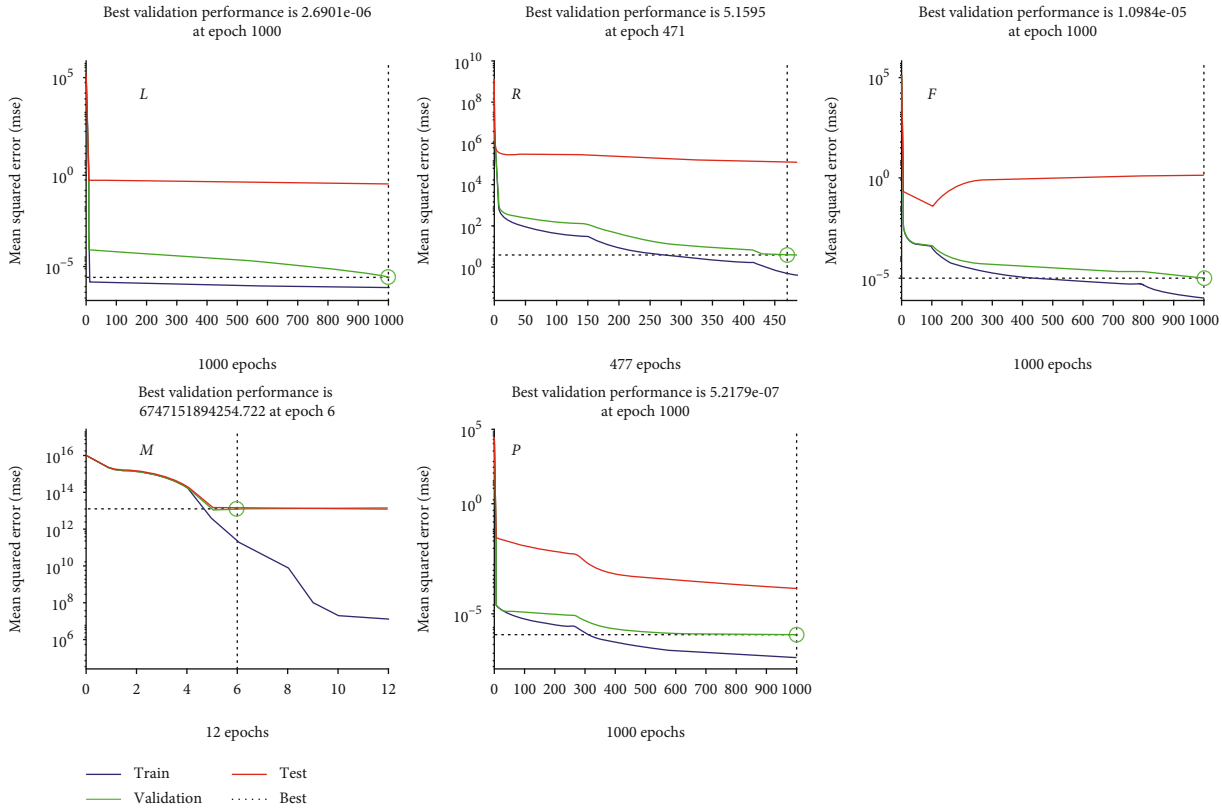


Figure 9: The graph of the neural network performance of model four.

Table 6: Model four neural network performance output.

| | $L$ | $R$ | $F$ | $M$ | $P$ |
|---|---|---|---|---|---|
| 20-layer neural network | 2.6901$e$-06 | 5.1595 | 1.0984$e$-05 | 6747151894254.722 | 5.2179$e$-07 |

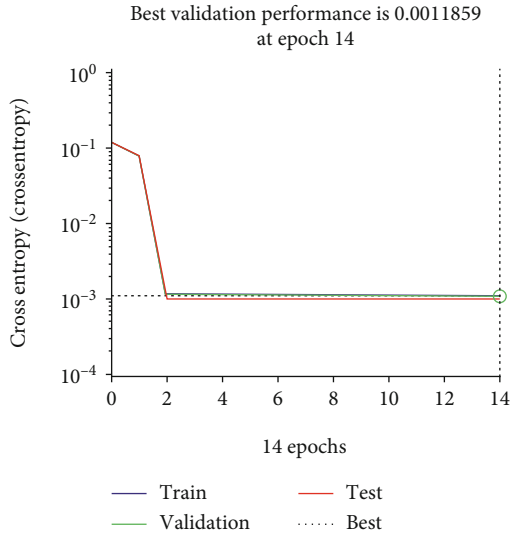


Figure 10: 20-layer neural network performance.

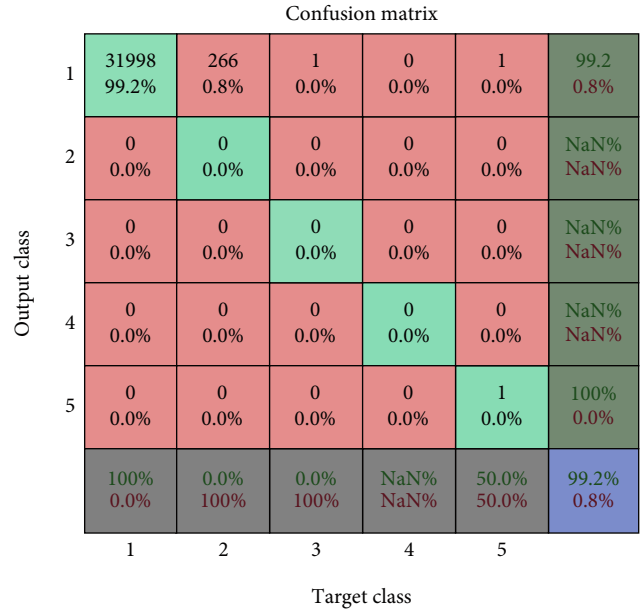


Figure 11: Collision matrix.

Table 7: Performance comparison of the mentioned models.

| | $L$ | $R$ | $F$ | $M$ | $P$ |
|---|---|---|---|---|---|
| 4-layerneural network | 9.1223$e$-05 | 0.02677 | 0.0010066 | 26283811960249.12 | 0.0037451 |
| 8-layerneural network | 0.33336 | 0.57701 | 3.0939$e$-05 | 170023661631120.31 | 0.00028733 |
| 10-layerneural network | 0.0016077 | 111.1271 | 32.2353 | 4236216134394.074 | 9.201$e$-08 |
| 20-layerneural network | 2.6901$e$-06 | 5.1595 | 1.0984$e$-05 | 6747151894254.722 | 5.2179$e$-07 |

FN represents the number of samples that have been misdiagnosed. In other words, the number of people who turn away is not recognized as turning away.

The following equations can also be used to evaluate the performance of the proposed algorithm and method.

$$\text{FMeasure} = \frac{2N_{\text{TP}}}{2N_{\text{TP}} + N_{\text{FP}} + N_{\text{FN}}}. \quad (5)$$

Equation (6) is used to calculate accuracy plus sensitivity.

$$\text{Precision} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{false positivies}}. \quad (6)$$

Kappa in Equation (7) shows the accuracy of meeting the desired expectations.

$$\text{Kappa} = \frac{2(\text{N}_{\text{TP}}\text{N}_{\text{TN}} + \text{N}_{\text{FN}}\text{N}_{\text{FP}})}{(\text{N}_{\text{TP}} + \text{N}_{\text{FN}})(\text{N}_{\text{TN}} + \text{N}_{\text{FN}})(\text{N}_{\text{TN}} + \text{N}_{\text{FP}})(\text{N}_{\text{TP}} + \text{N}_{\text{FP}})}. \quad (7)$$

*6.1. Integrated Model of the LRFMP Neural Network.* To integrate the proposed model, the LRFMP dataset is applied to the neural network simultaneously. The purpose of this work is to calculate the final value of each customer and the amount of credibility of the model.

Figure 10 shows the function of the neural network based on three types of training, testing, and evaluation data.

In Figure 11, the collision matrix resulting from the application of the model of the neural network with 20 hidden layers is shown.

## 7. Performance Comparison

To evaluate the proposed model, in addition to the described evaluation model, the average squares of the error obtained from the neural network are used. The average square error for each square sample is the error between the desired output and the actual output, and then the average is taken. The lower the average, the better and more acceptable the results. The neural network is implemented based on 4 models of 4 layers, 8 layers, 10 layers, and 20 separate layers, and the LRFMP output set is applied to the neural network separately and integrated, and the results are presented.

Regarding the functions of different models of the hidden layers of the neural network, Table 7 shows the comparison of the outputs of these models.

As shown in Table 7, the 20-layer model in $L$ showed better performance than other models; in $R$, the 4-layer model had the best performance, as well as in the clustering of the $F$-model, and the 20-layer model showed better performance, compared to other models. In a separate clustering of $M$, the 10-layer model exhibits a better performance and finally provides better performance in the P-10 clustering. In the integrated model, considering that only the 20-layer model was used, as shown in the collision matrix, the neural network with 99.2% yields a correct diagnosis of the resulting clusters.

## 8. Conclusion

Diverting forecast is an important tool for companies to keep up with competition in the market. In retail, a dynamic definition of disconnection is needed to identify the exact customers. This study examines the composition of the neural network, a new strategy for prediction of the deviation. It was found that the neural network was able to identify the tendency among all members, which is in fact the key to the prediction of the divergence. These results suggest that the neural network designed for the CLV time series prediction is a good strategy for rotating prediction. The metaparameter optimization strategy is not so problematic, and further optimization may be valuable; although, it may also improve the performance of the neural network. However, it should be pointed out that although the proposed model is used to identify trends, it is also used to identify members who deviate from the particular tendency of a specified group. It was also found in this research that clustering by LRFMP can be used to conduct a more comprehensive analysis of customer rejection. This algorithm provides meaningful and valid categories and predicts that there are patterns for dividing the dataset. Due to the limited time frame, only one clustering solution was investigated. The evaluation results indicate that the LRFMP can be used to test the return prediction efficiencies without the need for a rule extraction algorithm. The result is a shorter implementation time and easier implementation. The results of this research will be based on different models that can be used in customer evaluation.

The proposed algorithm does not depend on the type of characteristics, and the convergence in the relationships is important; on the other hand, due to the fact that the population is constant, the calculation time is acceptable. By examining the results, a high accuracy of 90% indicates that the proposed algorithm is efficient in converting and clustering customers. Therefore, in future development, the results can be optimized and stagnated.

There are numerous potential strategies for how to use the neural network along with the value of customer life

based on the definition of rejection. To develop this research model, other RFM models such as LRFM can be used, and the LRFMC model can be used instead of the LRFMP model. In the case of a neural network, recurring or repeated neural networks can be used to achieve a better result.

## Data Availability

The data is available in http://recsyswiki.com/wiki/Grocery_shopping_datasets.

## Conflicts of Interest

The author(s) declare(s) that they have no conflicts of interest.

## References


[1] P. A. Sarvari, A. Ustundag, and H. Takci, "Performance evaluation of different customer segmentation approaches based on RFM and demographics analysis," *Kybernetes*, vol. 45, no. 7, pp. 1129–1157, 2016.

[2] H. H. "Sunny" Hu, C. T. Huang, and P. T. Chen, "Do reward programs truly build loyalty for lodging industry?," *International Journal of Hospitality Management*, vol. 29, no. 1, pp. 128–135, 2010.

[3] A. Afram, F. Janabi-Sharifi, A. S. Fung, and K. Raahemifar, "Artificial neural network (ANN) based model predictive control (MPC) and optimization of HVAC systems: A state of the art review and case study of a residential HVAC system," *Energy and Buildings*, vol. 141, pp. 96–113, 2017.

[4] H. Lu and J. C. C. Lin, "Predicting customer behavior in the market-space: a study of Rayport and Sviokla's framework," *Information & Management*, vol. 40, no. 1, pp. 1–10, 2002.

[5] T. F. Bahari and M. S. Elayidom, "An efficient CRM-data mining framework for the prediction of customer behaviour," *Procedia Computer Science*, vol. 46, pp. 725–731, 2015.

[6] A. Ahmad, M. Y. Hassan, M. P. Abdullah et al., "A review on applications of ANN and SVM for building electrical energy consumption forecasting," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 102–109, 2014.

[7] E. F. Can, A. Ezen-Can, and F. Can, "Multilingual sentiment analysis: an RNN-based framework for limited data," 2018, http://arxiv.org/abs/1806.0451.

[8] A. Parvaneh, M. Tarokh, and H. Abbasimehr, "Combining data mining and group decision making in retailer segmentation based on LRFMP variables," *International Journal of Industrial Engineering & Production Research*, vol. 25, no. 3, pp. 197–206, 2014.

[9] S. M. S. Hosseini, A. Maleki, and M. R. Gholamian, "Cluster analysis using data mining approach to develop CRM methodology to assess the customer loyalty," *Expert Systems with Applications*, vol. 37, no. 7, pp. 5259–5264, 2010.

[10] D. C. Li, W. L. Dai, and W. T. Tseng, "A two-stage clustering method to analyze customer characteristics to build discriminative customer management: a case of textile manufacturing business," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7186–7191, 2011.

[11] J. T. Wei, S. Y. Lin, C. C. Weng, and H. H. Wu, "A case study of applying LRFM model in market segmentation of a children's dental clinic," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5529–5533, 2012.

[12] W. J. Reinartz and V. Kumar, "On the profitability of long-life customers in a noncontractual setting: an empirical investigation and implications for marketing," *Journal of Marketing*, vol. 64, no. 4, pp. 17–35, 2000.

[13] H. Chang and S. Tsay, "Integrating of SOM and K-mean in data mining clustering: An empirical study of CRM and profitability evaluation," *Journal of Information Management*, vol. 11, pp. 161–203, 2004.

[14] J.-T. Wei, S.-Y. Lin, and H. H. Wu, "A review of the application of RFM model," *African Journal of Business Management*, vol. 4, no. 19, pp. 4199–4206, 2010.

[15] R. Colombo and W. Jiang, "A stochastic RFM model," *Journal of Interactive Marketing*, vol. 13, no. 3, pp. 2–12, 1999.

[16] Y. Xu, R. Goedegebuure, and B. Van der Heijden, "Customer perception, customer satisfaction, and customer loyalty within Chinese securities business: towards a mediation model for predicting customer behavior," *Journal of Relationship Marketing*, vol. 5, no. 4, pp. 79–104, 2007.

[17] S. Peker, A. Kocyigit, and P. E. Eren, "LRFMP model for customer segmentation in the grocery retail industry: a case study," *Marketing Intelligence & Planning*, vol. 35, no. 4, 2017.

[18] S. Peker, A. Kocyigit, and P. E. J. K. Eren, "A hybrid approach for predicting customers' individual purchase behavior," *Kybernetes*, vol. 46, no. 10, pp. 1614–1631, 2017.

[19] K. Chen, Y. H. Hu, and Y. C. Hsieh, "Predicting customer churn from valuable B2B customers in the logistics industry: a case study," *Information Systems and e-Business Management*, vol. 13, no. 3, pp. 475–494, 2015.

Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8829868, 8 pages
https://doi.org/10.1155/2020/8829868

*Research Article*

# LSEA: Software-Defined Networking-Based QoS-Aware Routing Mechanism for Live-Soccer Event Applications in Smart Cities

**Yingcheng Zhang**[1] **and Gang Zhao**[2]

[1]*Northeast Normal University, Changchun 130024, China*
[2]*Shenzhen University, Shenzhen 518060, China*

Correspondence should be addressed to Gang Zhao; spjava111@163.com

Received 26 August 2020; Revised 6 October 2020; Accepted 11 October 2020; Published 7 November 2020

Academic Editor: Jianhui Lv



The smart cities provide a better connection between services and citizens based on new Internet technologies. During the building process of smart cities, some burgeoning applications have been emerging and changing the daily lifestyle of people, e.g., live streaming applications. Especially, the live-soccer event applications have attracted much attention and can improve people's enjoyment of life to a great extent, such as the Europe five major league matches and FIFA world cup. For such applications, the traditional routing strategies cannot do Quality-of-Service (QoS) awareness, and thus, the network performance and the Quality of Experience (QoE) of users cannot be guaranteed. In this paper, we employ Software-Defined Networking (SDN) to make QoS awareness for the special live-soccer event applications, in which the QoS-aware routing mechanism is proposed, called LSEA. Meanwhile, delay, delay jitter, and packet loss rate are considered as three objects. On this basis, the improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm are devised. Finally, the proposed LSEA is implemented over Mininet, and the experimental results demonstrate its feasibility and efficiency.

## 1. Introduction

With the rapid development of new Internet technologies such as mobile edge computing [1], 5G [2], and Internet of Things (IoT) [3], the building of smart cities [4, 5] has attracted much attention from many countries and regions, which can provide a better connection between services and citizens based on the combination of these new Internet technologies. At the same time, some burgeoning applications have been emerged and changed people's daily life patterns. For example, there have been a lot of live-broadcasting platforms [6], such as TikTok, Livestream, Twitch, and YouTube, and these platforms are used to spread some popular programmes. In particular, the live-soccer event applications [7] account for the large proportion of improving the people's enjoyment of life to a great extent, and the classical and well-known representatives are the Europe five major league matches [8] and FIFA world cup [9]. For the network in the smart cities, it is very necessary to recognize the live-soccer event application, which can enhance the network performance and improve the watching quality. However, the traditional Internet only provides the best-effort services and cannot do the differentiated scheduling for the special live-soccer event applications on the condition where the network resources in smart cities are limited [10]. On the other hand, the live-soccer events belong to the real-time applications, which have high requirements on delay, delay jitter, and packet loss rate. Since different applications have different requirements on Quality of Services (QoS) [11, 12] (for example, email applications have high requirements on packet loss rate and low requirements on delay and delay jitter), doing the application awareness and further making the differentiated scheduling are very significant. In other words, for the live-soccer event applications, they should be recognized in advance.

Furthermore, from the perspective of data transmission, the QoS-aware routing mechanism can be employed to provide differentiated services rather than the best-effort services [13]. In order to guarantee the QoS requirements of applications (e.g., the live-soccer events) during the process of data

transmission, the Internet Engineering Task Force (IETF) gives three well-known models, i.e., Integrated Services (IntServ) [14], Differentiated Services (DiffServ) [15], and Multi-Protocol Label Switching (MPLS) [16]. Among them, the IntServ model uses resource reservation protocol to ensure that each node has the reserved resources, that is to say, all nodes have to store the network status related to the services. However, the functions of different devices are different (difficult to unify these configurations of devices), which causes bad scalability and high complexity. Different from IntServ, DiffServ divides network applications into several service ranks, and the different service rank has different handling method. Especially when the phenomena of network congestion happens, the related data traffic will be scheduled according to the service rank. However, DiffServ only guarantees the QoS requirements of applications for each service rank but cannot for the end-to-end QoS guarantee. In the MPLS model, the node forwards data according to the label. Although this model is simple, it is very hard to configure, manage, and debug for the involved devices.

With the above statements, the Software-Defined Networking (SDN) [17–19] has been accepted as a novel network paradigm and it is a feasible and efficient solution to address QoS awareness for the special live-soccer event applications. Different from the traditional network architecture, SDN separates the data plane from the control plane, i.e., managing the network system via the concentrated way. In addition, SDN can easily obtain the global network view and status information, such as flow statistics, the availability of network resources, the network topology information, and even the more detailed forwarding/routing information. On this basis, the control plane of SDN gives real-time control policies, and then, they are devolved to the data plane, which can guarantee the unified management and configuration for the network resources. In particular, the control plane of SDN can customize different control policies for different data flow according to the QoS requirements, in order to complete the fine-grained management. In other words, SDN can provide the differentiated end-to-end services for the special live-soccer event applications, greatly enhancing the network performance and improving the watching quality.

Therefore, we in this paper plan to employ SDN to make QoS awareness for the special live-soccer event applications in smart cities, in which the QoS-aware routing mechanism is proposed, called LSEA. The main contributions of this paper are summarized as follows. (1) SDN-based QoS-aware routing model is introduced, where delay, delay jitter, and packet loss rate are considered as three objects for optimization. (2) The improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm are devised for the differentiated end-to-end services. (3) The proposed LSEA is implemented over Mininet to prove its feasibility and efficiency based on four metrics, i.e., delay, delay jitter, packet loss rate, and system recovery time.

The rest of the paper is organized as follows. Section 2 reviews the related work. In Section 3, the system model and three objective functions are quantified. Section 4 introduces the method of network status information collection. Section 5 devises the improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm. Section 6 reports the experimental results, and finally, Section 7 concludes this paper.

## 2. Related Work

As we know, SDN-based QoS-aware routing mechanisms mainly include a single-path-based routing mechanism and a multiple-path-based routing mechanism.

There have been some single-path-based routing mechanisms guaranteeing the QoS requirements of applications. In [20], the OpenFlow switches were enabled to cooperate with the legacy switches by using the learning bridge protocol without requiring any modification on legacy switches. By utilizing the characteristics of SDN, SDN applications could dynamically find routing paths according to predefined QoS requirements and current network status. In [21], the use of QoS-based routing scheme over SDN was investigated, in which a real SDN test-bed is constructed by using Raspberry Pi computers as virtual SDN switches managed by a centralized controller. In [22], the model of adaptive routing of heterogeneous traffic with respect to the current QoS provisioning requirements was proposed, of which the main idea was to develop the model for effective routing with a high degree of flexibility achieved by using a set of weighting coefficients which all together constituted the general routing metrics. In [23], a server-driven bit rate estimation approach to compute the video bit rate and inform the application QoS requirement to the control layer was proposed. In addition, a QoS routing design for adaptive stream was devised, which allowed the SDN controller to evaluate all passable paths based on whole network topology by taking the bit rate of the segments into account.

Furthermore, some multiple-path-based routing mechanisms to guarantee the QoS requirements of applications have also been proposed. In [24], a new routing algorithm to calculate the bandwidth-delay constrained routes in a fast and efficient manner was presented. The algorithm was designed for the software-defined backbone networks, where the control plane was separated from the data plane and logically centralized. Besides providing the required QoS, the algorithm is aimed at maximizing the utilization of network resources. In [25], a source routing scheme to conduct the top-K QoS-aware paths discovery in SDN was introduced. First, the novel noninvasive QoS scheme was designed to collect QoS information based on LLDP in a piggyback fashion. Then, the variations of the K-shortest path algorithm were derived to find the unconstrained/constrained top-K ranked paths with regard to individual/overall path costs, reflecting QoS. In [26], the utilization of segment routing in SDN-based networks was explored to improve the network resource utilization and end users' QoS for delivering multimedia services over 5G networks. In [27], a mechanism to support on-demand QoS routing without any help from traffic engineering in SDN was proposed. By exploiting the fact that a controller could provide multiple routing paths based on global topology in SDN, the authors further proposed to monitor QoS over multiple routing paths and select the appropriate path satisfying users' QoS requirement. In [28], the end-to-end QoS based on the queue support in

OpenFlow was proposed, which allowed an operator with an SDN-enabled network to efficiently allocate the network resources according to the users' demands. Especially for each flow, the proposed solution guaranteed the required end-to-end QoS, while efficiently managing the utilization of open virtual switches.

As a matter of fact, the single-path routing can satisfy the QoS requirements due to the regulation of multiple weight factors. However, when the given path happens network congestion, it cannot provide the subsequent QoS provisioning services. Thus, in order to guarantee the reliability of routing, the multiple-path routing has been investigated. However, when the congestion phenomena of the common link(s) happen, a similar problem with respect to the single-path routing will emerge. Given such consideration, this paper plans to improve the Dijkstra routing algorithm and devise the SDN-based disjoint routing algorithm, guaranteeing the differentiated end-to-end services for the special live-soccer event applications.

## 3. Frame Structure

*3.1. System Model.* As depicted in Figure 1, the proposed LSEA is consisted of two major modules, i.e., network status information collection and path computing for the live-soccer events. Therein, the network status information collection module has two functions, i.e., topology management and network measurement, which is completed by the OpenFlow switch. The path computing module includes the improved Dijkstra routing and the SDN-based disjoint routing, which is used for the differentiated scheduling with respect to the special live-soccer event applications. In particular, the improved Dijkstra routing is used to obtain the multiple equivalent shortest paths in order to improve the response speed. The SDN-based disjoint routing is aimed at determining some disjoint paths from those equivalent shortest paths in order to improve the fault tolerance and the reliability of QoS routing. Meanwhile, the optimal path from these disjoint paths is selected to transmit the application data related to live-soccer events. To sum up, the improved Dijkstra routing is the foundation of SDN-based disjoint routing.

*3.2. Optimization Objective.* In order to learn the network status, more network information metrics should be collected. Based on the adequate information collection, the proper routing decision can be made well. Furthermore, in order to satisfy the various service requirements, this paper computes QoS routing with the multiple factors considered, balancing different factors. In particular, the multiple factors are originated from QoS parameters, including delay, delay jitter, and packet loss rate. However, the multifactor constraint QoS routing belongs to the NP-hard problem. Given this, we adopt the weighted evaluation function to transform the multiobjective model into a single-objective model, structuring a new routing measurement. In addition, different applications have different QoS requirements; thus, this paper adjusts the delay, delay jitter, and packet loss rate to change the objective function value and further satisfy the different QoS requirements. For example, email applications are impacted by the packet loss rate. We can adjust the weight value with respect to the packet loss rate to be large, so that these links with a high packet loss rate can be bypassed effectively.

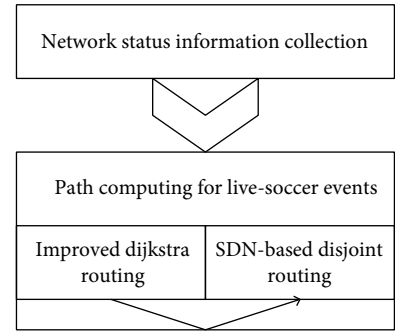


Figure 1: The frame structure of LSEA.

Given two arbitrary nodes $R_i$ and $R_j$, let $de_{ij}$, $jt_{ij}$, and $ls_{ij}$ denote the delay, the delay jitter, and the packet loss rate between $R_i$ and $R_j$, respectively, and the single-objective function is defined as follows.

$$\text{minimize } f_{ij} = \alpha * de_{ij} + \beta * jt_{ij} + \gamma * ls_{ij}, \tag{1}$$

where $\alpha$, $\beta$, and $\gamma$ are the weights with respect to delay, delay jitter, and packet loss rate, respectively, and $\alpha + \beta + \gamma = 1$. However, the three metrics are not in the same order of magnitude; thus, the standardization operation should be performed.

Let $de'_{ij}$, $jt'_{ij}$, and $ls'_{ij}$ denote the standardized results on $de_{ij}$, $jt_{ij}$, and $ls_{ij}$, respectively, and we have

$$de'_{ij} = \frac{de_{ij} - de_{\min}}{de_{\max} - de_{\min}}, \tag{2}$$

where $de_{\max}$ and $de_{\min}$ are the maximal and minimal link delay between $R_i$ and $R_j$.

$$jt'_{ij} = \frac{jt_{ij} - jt_{\min}}{jt_{\max} - jt_{\min}}, \tag{3}$$

where $jt_{\max}$ and $jt_{\min}$ are the maximal and minimal link delay jitter between $R_i$ and $R_j$.

$$ls'_{ij} = \frac{ls_{ij} - ls_{\min}}{ls_{\max} - ls_{\min}}, \tag{4}$$

where $ls_{\max}$ and $ls_{\min}$ are the maximal and minimal link packet loss rate between $R_i$ and $R_j$.

## 4. Network Status Collection

The traditional network status information collection usually adopts the offline method since the network cannot automatically obtain the traffic information. As a result, it is very necessary to find an online method to complete the efficient QoS-aware routing for the special live-soccer event

applications while there is no need to spend much time for collecting the statistical information. With such consideration, we use SDN to collect the network status information for all applications, because SDN has a global awareness function on the network status information without the additional overhead.

In terms of the network status information collection in SDN, some flow tables are requisite. However, these flow tables are only installed by the OpenFlow protocol [29, 30]; therefore, SDN needs the help of the OpenFlow switch, that is to say, this paper deploys OpenFlow switch to help complete the network status information collection under SDN environment.

Based on the above discussion, the structure of network status information collection is shown in Figure 2, including the SDN network environment and the traditional Internet environment. Meanwhile, the communication between the SDN network and the traditional Internet depends on two switches: ordinary switch and OpenFlow switch. Regarding the network status information collection based on two switches, the related process is described as follows. At first, the ordinary switch collects the network status information of all users. Then, the ordinary switch submits the collected network status information to the OpenFlow switch by the techniques of port mirroring and redirection, just like the data replication. Finally, the OpenFlow switch sends the related network status information to its corresponding SDN controller.

## 5. QoS-Aware Path Computing

This section will make the path computing, i.e., realizing the QoS-aware routing mechanism for the special live-soccer event applications in smart cities, including the improved Dijkstra routing and the SDN-based disjoint routing. Meanwhile, the former is used to obtain the multiple equivalent shortest paths in order to improve the response speed, while the latter is to select two or three disjoint paths from those equivalent shortest paths in order to improve the fault tolerance and the reliability of QoS routing.

*5.1. Improved Dijkstra Routing.* The original Dijkstra algorithm computes all shortest paths with respect to any two nodes, which cannot satisfy the requirements of allocating different QoS ranks for different applications because of two major limitations, as follows.

(i) It computes all shortest paths regarding any two nodes, which increases the time computation overhead

(ii) It only computes the shortest path between any two nodes, irrespective of the computing of multiple shortest paths between any two nodes, because the data structure to store the front node is only an array (only record a front node in terms of the current node) and cannot store the multifront nodes.

In order to satisfy the requirements of different QoS rank allocation, the multiple equivalent shortest paths should be selected in advance. Given this, we improve the Dijkstra routing algorithm based on the following two aspects.

(i) When the system finds the destination node, the routing algorithm is finished, which can decrease the time computation overhead.

(ii) The data structure of storing the front node is modified from the array to the linked list.

Based on the above statements, the improved Dijkstra routing algorithm is described in Algorithm 1, in which the new measurement value on the link is updated according to equation (1).

Among them, $T_$nodes is the total number of nodes, pre_nodes is the set of front nodes regarding the current node, dist is the distance between source node to the destination node, detec is the set of nodes which have been detected, and undetec is the set of nodes which have not been detected.

*5.2. SDN-Based Disjoint Routing.* For one path, the smaller remaining bandwidth means the larger probability to happen network congestion. Here, the remaining bandwidth of the path is defined as the minimal bandwidth value among all involved links with respect to the current path. As the above mentioned, SDN can have a good command of knowledge on the global network view. Therefore, the SDN controller can measure the remaining bandwidth information periodically. Furthermore, in order to avoid network congestion, this paper considers the path that has the maximal remaining bandwidth as the optimal path to transmit the live-soccer event applications, which can guarantee the QoS requirements, improve the throughput and response time, and complete the load balance.

Moreover, some disjoint paths from those equivalent shortest paths obtained by the improved Dijkstra routing algorithm are selected to guarantee the fault tolerance and the reliability of QoS routing, where the remaining bandwidth of the path is regarded as the evaluation metric. In particular, the path with the maximal remaining bandwidth is used for the main routing, while the rest of the paths are used for the alternative routing.

For example, in Figure 3, the cost of the link refers to the bandwidth. Given two paths from node A to D, i.e., A-B-D and A-B–C-D. The first path's remaining bandwidth is 1 (min $\{3, 1\}$), and the second path's remaining bandwidth is 2 (min $\{3, 4, 2\}$). As a conclusion, the second path is regarded the main routing.

The SDN-based disjoint routing is composed of the following four operations. At first, the path is converted into some links. Then, according to the link's remaining bandwidth information, the path's remaining bandwidth is updated through the form of iteration. Thirdly, these paths are arranged in the descending order according to the path's remaining bandwidth. Finally, the path with the maximal remaining bandwidth is determined in advance, and the correspondingly involved links constitute some disjoint paths. Based on the above statements, the detailed SDN-based disjoint routing algorithm is described in Algorithm 2.

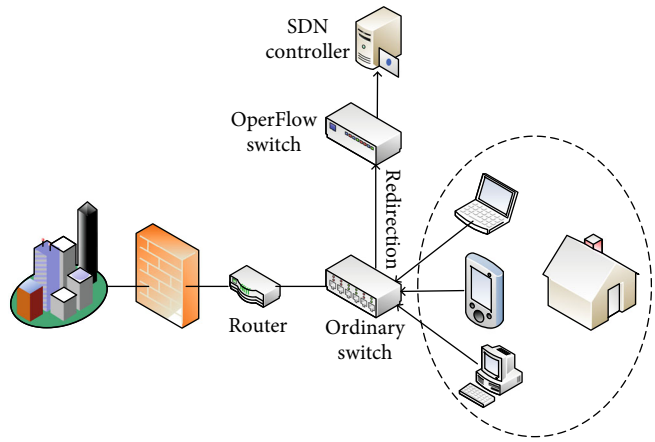


Figure 2: The structure of network status information collection.

01: Initialize $T$_nodes, pre_nodes, dist, detec, undetec, and Path;
02: Update the new measurement value according to equation (1);
03: Store the sour to detec;
04: **while** undetec ≠ Null and node ≠ dest, **do**
05: Select node according to the shortest dist;
06: Update dist;
07: Update pre_nodes;
08: **endwhile**
09: Compute the multiple equivalent shortest paths according to pre_nodes;
10: Update Path;
01: **return** Path;

Algorithm 1: The improved Dijkstra routing.

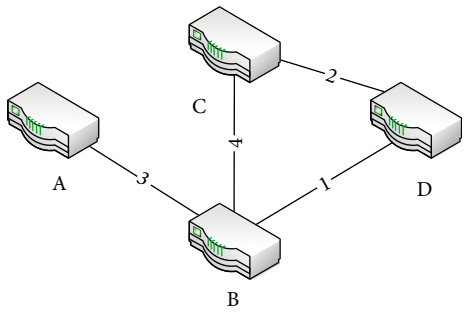


Figure 3: An example to illustrate the determination of the main routing.

Among them, minbw is the set used to store the minimal remaining bandwidth for each path, $P$_to_$L$ is the set used to store all involved links in terms of the current path; disjP is the set used to store the obtained disjoint paths, and pri_ min _bw is the minimal remaining bandwidth to which disjP corresponds.

Particularly, from the perspective of time complexity, Algorithm 1 runs in $O(n^2 \log n)$ because it only computes the shortest path between the source node and destination node; Algorithm 2 runs in $O(n \log n)$ because SDN has the global network view and the main computation overhead comes from the arrangement operation.

## 6. LSEA Evaluation

*6.1. Environment Setup.* The proposed LSEA in smart cities is implemented based on the Intel(R) Core(TM) i5-8500 CPU @3.00GHz, RAM 8.00GB, running on the Ubuntul6.02 64bits operation systems. The programming language is Python, running on the Pycharm. The SDN network environment is implemented based on Mininet [31], where the Ryu controller is employed and the Iperf instrument is used to send packets. The simple process is described as follows. At first, the SDN network environment is created over Mininet; secondly, the network environment is connected with the remote Ryu controller; thirdly, two hosts are opened: one is regarded as the source and the other one is regarded as the destination, in which the Iperf instrument is started to send packets; finally, the path is computed to forward the live-soccer event applications.

*6.2. Data Collection.* At present, there are no open datasets on delay, delay jitter, loss packet rate, and bandwidth information; thus, this paper uses some programs to simulate the similar dataset information, including QoS parameters. At first, we use files to store the network topology: if two nodes are adjacent, the corresponding status is expressed by 1; otherwise, the corresponding status is expressed by 0. Then, the file of network topology is read, generating the related QoS information. Finally, three metrics, i.e., delay, delay

```
01: Initialize minbw, P_to_L, disjP and pri_ min _bw;
02: Put Path into P_to_L;
03: For each path in Path, do
04: Update minbw;
05: minbw = newminbw;
06 endfor
07: Arrange these paths in Path according to minbw;
08: Compute disjP according to minbw;
09: Obtain pri_ min _bw according to disjP;
10: return disjP;
```

Algorithm 2: The SDN-based disjoint routing.

jitter, and loss packet rate, are standardized to guarantee that they are at the same order of magnitude.

*6.3. Simulation Setting.* The Deltacom network topology with 97 nodes and 124 links is used for simulation, as shown in Figure 4, where there are three requesters and four providers. The delay, delay jitter, loss packet rate, and system recovery time are considered as four performance evaluation metrics. In addition, two baselines, i.e., single-path-based QoS routing mechanism [23] and multiple-path-based QoS routing mechanism [27], shorten for ISPCS and NSW, respectively, are selected as the comparison benchmarks. Three parameters, $\alpha$, $\beta$, and $\gamma$, are randomly set as 0.3, 0.4, and 0.3, respectively. For each number of packets, we make 100 times simulations, that is to say, the average experiment results are based on 100 times simulations.

*6.4. Results*

*6.4.1. Delay.* The experimental results on delay are shown in Figure 5. We can find that the average delays of LSEA, ISPCS, and NSW are 80.26 ms, 89.38 ms, and 84.38 ms, respectively; that is to say, the proposed LSEA has the obvious advantage on the average delay. In fact, the weight of delay in LSEA is larger than those in both ISPCS and NSW, which can filter out some links with the relatively large delay. Thus, the end-to-end delay of LSEA is smaller than those of ISPCS and NSW.

*6.4.2. Delay Jitter.* The experimental results on delay jitter are shown in Figure 6. We can find that the average delay jitters of LSEA, ISPCS, and NSW are 10.63 ms, 13.17 ms, 15.12 ms, respectively. In other words, LSEA has smaller delay jitter than ISPCS and NSW; this is because LSEA filters out some links with the relatively large delay jitter via the setting of$\beta$. In addition, the SDN-based disjoint routing mechanism can guarantee the stable transmission of live-soccer event applications. However, ISPCS and NSW do not consider disjoint routing.

*6.4.3. Loss Packet Rate.* The experimental results on the loss packet rate are shown in Figure 7. We can find the average loss packet rates of LSEA, ISPCS, and NSW are 0.042, 0.038, and 0.041, respectively. Although the proposed LSEA has a relatively larger loss packet rate than ISPCS and NSW, the corresponding results are very close. Based on such consideration, we think that the proposed LSEA is acceptable in terms of the loss packet rate.

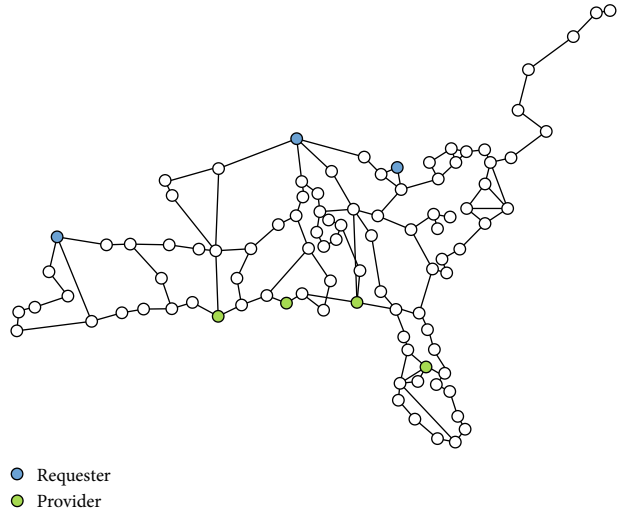


Figure 4: Deltacom network topology used for simulation.

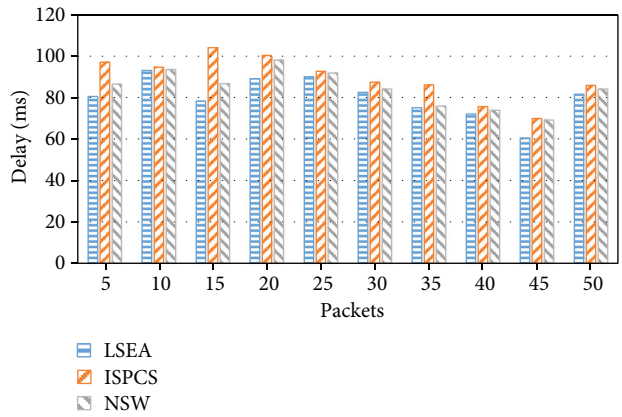


Figure 5: The experimental results on the average delay.

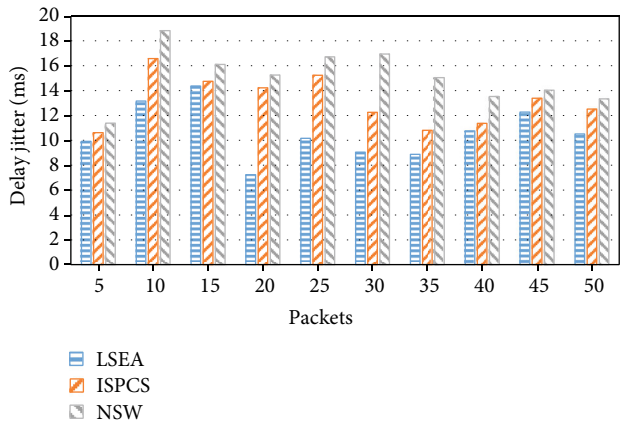


Figure 6: The experimental results on the average delay jitter.

*6.4.4. System Recovery Time.* In this section, we disconnect two or three links periodically, and the time period is set as 5 s. The experimental results on system recovery time are shown in Figure 8. We can find that the proposed LSEA

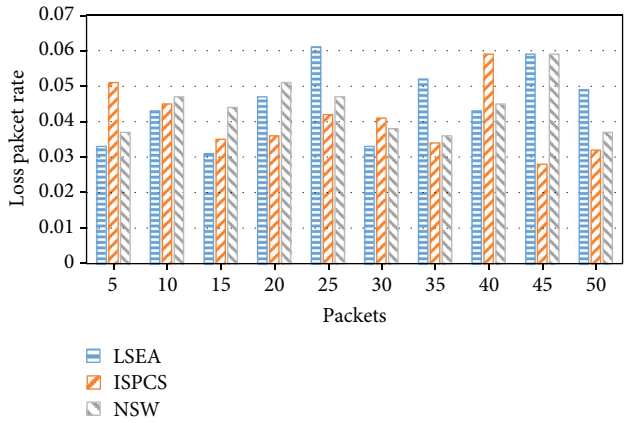


Figure 7: The experimental results on the average loss packet rate.

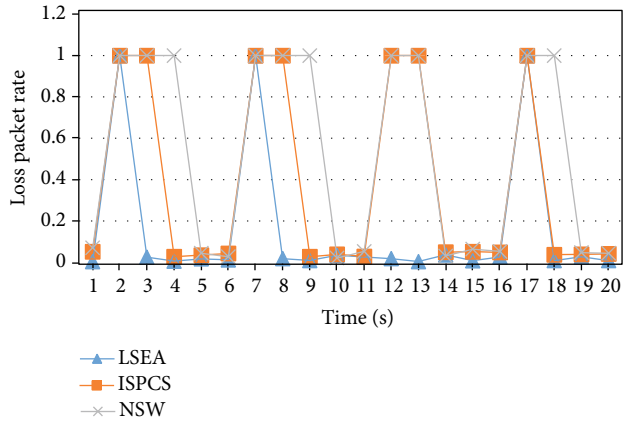


Figure 8: The experimental results on the average system recovery time.

always can recover the system stability with the fastest speed. Two reasons are listed as follows. On the one hand, LSEA has some disjoint paths, which can provide the transmission guaranteeing for the special live-soccer event applications. On the other hand, LSEA uses SDN to control the network in a centralized manner, which can schedule the feasible path for the live-soccer event applications in case of link failure.

## 7. Conclusions

With the rapid development of new Internet technologies such as mobile edge computing, 5G, and IoT, the building of smart cities has attracted much attention from many countries and regions, which can provide a better connection between services and citizens based on the combination of these new Internet technologies. For the special live-soccer event applications in the smart cities, this paper employs SDN to make QoS-awareness routing. At first, the SDN-based QoS-aware routing model is introduced, where delay, delay jitter, and packet loss rate are considered as three objects for optimization. Then, the improved Dijkstra routing algorithm and SDN-based disjoint routing algorithm are devised for the differentiated end-to-end services. In particular, the improved Dijkstra routing is the foundation of SDN-based disjoint routing. Finally, the proposed LSEA is implemented over Mininet to prove its feasibility and efficiency based on delay, delay jitter, packet loss rate, and system recovery time.

However, this paper also has some limitations. On the one hand, the analysis of time complexity and space complexity is not included. On the other hand, more applications should be employed to make the simulation experiments. In the future, we will do a further study on LSEA around the abovementioned two limitations. Furthermore, we also plan to improve LSEA based on the method of machine learning.

## Abbreviations

DiffServ: Differentiated services
IETF: Internet Engineering Task Force
IntServ: Integrated Services
IoT: Internet of Things
ISPCS: Intelligent signal processing and communication systems [23]
LSEA: The proposed routing mechanism in this paper
MPLS: Multi-Protocol Label Switching
NSW: Network softwarization and workshops [27]
QoS: Quality of Service
SDN: Software-defined networking.

## Data Availability

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

## Conflicts of Interest

None of the authors have any conflicts of interest.

## Acknowledgments

We thank Ping Li, who is a senior experimenter to provide the SDN network environment.

## References

[1] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: progress and challenges," in *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, pp. 83-84, Oxford, UK, March 2016.

[2] P. T. Dat, A. Kanno, N. Yamamoto, and T. Kawanishi, "5G transport networks: the need for new technologies and standards," *IEEE Communications Magazine*, vol. 54, no. 9, pp. 18–26, 2016.

[3] Y. Liu, K. Wang, K. Qian, M. Du, and S. Guo, "Tornado: enabling blockchain in heterogeneous Internet of Things through a space-structured approach," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1273–1286, 2020.

[4] S. Sengupta and S. S. Bhunia, "Secure data management in cloudlet assisted IoT enabled e-Health framework in smart city," *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9581–9588, 2020.

[5] E. J. Cedillo-Elias, J. A. Orizaga-Trejo, V. M. Larios, and L. A. M. Arellano, "Smart government infrastructure based in SDN networks: the case of guadalajara metropolitan area," in *2018*

*IEEE International Smart Cities Conference (ISC2)*, pp. 1–4, Kansas City, MO, USA, September 2018.

[6] Y. Li, W. Ren, T. Zhu, Y. Ren, Y. Qin, and W. Jie, "RIMS: a real-time and intelligent monitoring system for live-broadcasting platforms," *Future Generation Computer Systems*, vol. 87, pp. 259–266, 2018.

[7] X. Yu, L. Li, and H. W. Leong, "Interactive broadcast services for live soccer video based on instant semantics acquisition," *Journal of Visual Communication and Image Representation*, vol. 20, no. 2, pp. 117–130, 2009.

[8] V. Hofer and J. Leitner, "Relative pricing of binary options in live soccer betting markets," *Journal of Economic Dynamics and Control*, vol. 76, pp. 66–85, 2017.

[9] J. L. Nicolau and A. Sharma, "A generalization of the FIFA World Cup effect," *Tourism Management*, vol. 66, pp. 315–317, 2018.

[10] U. Gulec, M. Yilmaz, V. Isler, R. V. O'Connor, and P. M. Clarke, "A 3D virtual environment for training soccer referees," *Computer Standards & Interfaces*, vol. 64, pp. 1–10, 2019.

[11] C. Abid, M. Kessentini, and H. Wang, "Early prediction of quality of service using interface-level metrics, code-level metrics, and antipatterns," *Information and Software Technology*, vol. 126, article 106313, 2020.

[12] J. O. Mebawondu, F. M. Dahunsi, and O. S. Adewale, "Hybrid intelligent model for real time assessment of voice quality of service," *Scientific African*, vol. 9, article e00491, 2020.

[13] Q. He, J. Yan, H. Jin, and Y. Yang, "Quality-aware service selection for service-based systems based on iterative multi-attribute combinatorial auction," *IEEE Transactions on Software Engineering*, vol. 40, no. 2, pp. 192–215, 2014.

[14] Z. Shan, "Integrated service adaptation," in *2010 6th World Congress on Services*, pp. 140–143, Miami, FL, USA, July 2010.

[15] Y. Wang, X. Wang, H. Li, Y. Dong, Q. Liu, and X. Shi, "A multi-service differentiation traffic management strategy in SDN cloud data center," *Computer Networks*, vol. 171, article 107143, 2020.

[16] Z. Al-Qudah, I. Jomhawy, M. Alsarayreh, and M. Rabinovich, "On the stability and diversity of Internet routes in the MPLS era," *Performance Evaluation*, vol. 138, article 102084, 2020.

[17] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): a survey," *Journal of Network and Computer Applications*, vol. 80, pp. 200–218, 2017.

[18] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, article 106984, 2020.

[19] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: a survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.

[20] C. Lin, K. Wang, and G. Deng, "A QoS-aware routing in SDN hybrid networks," *Procedia Computer Science*, vol. 110, pp. 242–249, 2017.

[21] A. Kucminski, A. Al-Jawad, P. Shah, and R. Trestian, "QoS-based routing over software defined networks," in *2017 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pp. 1–6, Cagliari, Italy, June 2017.

[22] M. Beshley, M. Seliuchenko, O. Panchenko, and A. Polishuk, "Adaptive flow routing model in SDN," in *2017 14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, pp. 298–302, Lviv, Ukraine, 2017.

[23] X. Jin, H. Ju, S. Cho, B. Mun, C. Kim, and S. Han, "QoS routing design for adaptive streaming in software defined network," in *2016 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 1–6, Phuket, Thailand, October 2017.

[24] S. Tomovic and I. Radusinovic, "Fast and efficient bandwidth-delay constrained routing algorithm for SDN networks," in *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 303–311, Seoul, South Korea, June 2016.

[25] X. Chen, J. Wu, and T. Wu, "The top-K QoS-aware paths discovery for source routing in SDN," *KSII Transactions on Internet & Information Systems*, vol. 12, no. 6, pp. 2534–2553, 2018.

[26] A. Barakabitze, L. Sun, I. Mkwawa, and E. Ifeachor, "A novel QoE-centric SDN-based multipath routing approach for multimedia services over 5G networks," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, Kansas City, MO, USA, May 2018.

[27] T. Kim and T. Nguyen-Duc, "OQR: on-demand QoS routing without traffic engineering in software defined networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, pp. 362–365, Montreal, QC, Canada, June 2018.

[28] D. L. C. Dutra, M. Bagaa, T. Taleb, and K. Samdanis, "Ensuring end-to-end QoS based on multi-paths routing using SDN technology," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Singapore, Singapore, December 2017.

[29] E. L. Fernandes, E. Rojas, J. Alvarez-Horcajo et al., "The road to BOFUSS: the basic OpenFlow userspace software switch," *Journal of Network and Computer Applications*, vol. 165, article 102685, 2020.

[30] R. Jmal and L. C. Fourati, "An OpenFlow architecture for managing content-centric-network (OFAM-CCN) based on popularity caching strategy," *Computer Standards & Interfaces*, vol. 51, pp. 22–29, 2017.

[31] Mininethttp://mininet.org.