Computer Networks 185 (2021) 107709

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A QoS-guaranteed intelligent routing mechanism in software-defined networks

Weifeng Sun *, Zun Wang, Guanghao Zhang

Key Lab Intelligent Control & Optimizat Ind Equip, Dalian University of Technology, Dalian, China

ARTICLE INFO

Keywords:
SDN
IoT
Data flow classification
QoS guaranteed routing

ABSTRACT

With the development of the Internet of Things (IoT), the network is required to guarantee the differential Quality of Service (QoS) requirements of the various data flows of various IoT services. Software-defined network (SDN) is envisioned as a promising technique to guarantee the QoS requirements of different services, through separating the control logic from data planes of networks. In order to guarantee the QoS requirements of data flows in SDNs, in this paper we investigate the problem of intelligent routing in SDNs, by leveraging a novel data flow classification method. Combining a variety of machine learning algorithms, a data flow classification method called MACCA2-RF&RF is proposed, in order to identify the data flow category and obtain the QoS requirements. The link parameter is newly designed considering multiple QoS requirements. According to the link parameter, the QoS-guaranteed path selection algorithm is then proposed, which can select QoS guaranteed routing path for different data flows with different QoS requirements. Aiming at the situation that the link is congested, local routing change algorithm is then proposed which only adjusts the links before and after the congested link instead of the entire path. Based on the above, a QoS-guaranteed intelligent routing mechanism called QI-RM in SDN is finally proposed in this paper to achieve QoS guarantee for data flows. The simulation results show that the MACCA2-RF&RF can classify data flows efficiently with an identification accuracy of 99.73%, and the QI-RM can guarantee the QoS requirements of data flows before and after link congestion.

1. Introduction

The Internet of Things (IoT) has become more and more widespread in daily life and production. The smart IoT that reforms the future production with the considerations of technological, economic, energy-efficient and social sustainability will become one of next promising research directions for both academia and industries [1]. The large-scale deployment and application of various sensors in IoT have generated a large amount of data [2]. For example, a large number of devices in the Industrial IoT (IIoT) applications produce numerous data with a large variety, which requires the networks to have a large data capacity and a high transmission speed [3].

According to the initial definition of ITU-T, quality of service (QoS) is a composite indicator, being used to assess user's satisfaction with a certain service. Generally speaking, QoS means the quality of the service provided by the network for specified data transmissions in a certain aspect. For example, if the network provides low-latency transmission services for delay-sensitive data transmission, QoS means low delay; if the network guarantees sufficient bandwidth for bandwidth-sensitive data transmission, QoS means bandwidth guaranteed. IoT is a composite system, which includes many functions and services, including perception, transmission, decision-making and so on. In addition, IoT contains various devices, including perception devices, communication networks, and intelligent data processing devices. QoS will exists in various data flows and the data flows of different services have different QoS requirements. For example, QoS of data flow used for inquiry generally focuses on reliability, as for QoS of data flow used for control focuses on the timeliness of the information. In addition to timeliness and reliability, QoS also involves bandwidth, packet loss rate, data accuracy, service level and so on. The QoS of different data flows in IoT should be guaranteed as much as possible, which means that the performance of each data flow is guaranteed, so that the overall performance of IoT can be guaranteed. So it is very important to guarantee the QoS requirements of the data flows of the IoT, so as to ensure the normal operation of the relevant services of the IoT for users.

Software-defined network (SDN) is an emerging network architecture, which separates the control and transmission functions of the network into the control plane and the data plane [4]. Traffic control

* Corresponding author.
E-mail addresses: wfsun@dlut.edu.cn (W. Sun), wangzun_ssdut2020@mail.dlut.edu.cn (Z. Wang), zhanggh5332@mail.dlut.edu.cn (G. Zhang).

https://doi.org/10.1016/j.comnet.2020.107709
Received 15 July 2020; Received in revised form 27 October 2020; Accepted 22 November 2020
Available online 4 December 2020

decisions are made by controllers in the control plane. The controller is programmable and can obtain the status of the entire network. The transmission of data flows is done by devices in the data plane. To guarantee QoS requirements of different data flows, it is necessary to solve the problems caused by unreasonable resource allocation in the network. The traditional networks are distributed, and each router cooperates to complete the data forwarding. It is difficult for traditional networks to achieve global and effective traffic control. Traditional service models (such as Best Effort, Int-Serv, Diffserv) have some problems and can hardly guarantee the QoS requirements of different data flows in IoT. For example, in Int-Serv, in order to obtain a certain quality of service, a data flow must request the network to reserve some resources for it before transferring data to the network. Switches need to maintain a large amount of state information, which is not conducive to the management and expansion of the network. But it is achievable for SDN. SDN controllers have a global vision, which is responsible for network monitoring, the protocol calculation and can provide data flow level control through centralized control [5]. SDN can provide data manipulation at the data flow level, it can better guarantee the QoS requirements of data flows than traditional network.

How to guarantee QoS requirements of data flows is a hot research issue in the field of SDN. For unknown data flows, the controllers need to obtain the QoS requirements in order to guarantee the QoS. In IoT, different types of data flows have different QoS requirements, which can be obtained by identifying the application category of the data flow. So an efficient and accurate classification algorithm is needed in SDN to quickly classify the data flows and obtain the QoS requirements. Different data flows have different requirements on the link state (remaining bandwidth, delay, packet loss rate and so on). So after obtaining the QoS requirements, the QoS guaranteed algorithms in SDN need to be able to select suitable paths for different data flows according to link states in order to guarantee QoS. When the link is congested, the QoS requirements of data flow may not be guaranteed, thus affecting the normal operation of relevant IoT services. Hence the QoS guarantee algorithm in SDN should also adjust the transmission path of the data flows in time.

In [6], we proposed a data flow classification algorithm (MACCA) for obtaining the QoS requirements of data flows in SDN. MACCA is composed of four modules, which are two base classifiers, a misclassification results judgment module and a decision module. MACCA is a combination of multiple machine learning methods, which can reduce the classification time while maintaining a high classification accuracy. In this paper, in order to improve MACCA, different machine learning methods have been tried to be applied to each module in MACCA, forming in a variety of improvements, among which, MACCA2-RF&RF has the highest accuracy rate and shorter classification time. Compared with MACCA, MACCA2-RF&RF replaces the C4.5 algorithm in the base classifiers and misclassification results judgment module with the Classification and Regression Tree (CART) algorithm and random forest (RF) algorithm.

Based on the MACCA2-RF&RF, different QoS thresholds and link parameter weights are set for different QoS requirements. In this paper, QoS means sufficient bandwidth, low delay, low packet loss rate and link load balancing. Load balancing can improve bandwidth utilization and reduce delay on the links. Therefore, the QoS requirement of load balancing also means the QoS requirements of bandwidth and delay. Then a QoS guaranteed path selection algorithm and a local routing change algorithm are newly proposed for QoS guaranteed routing before and after link congestion. Combining the MACCA-RF&RF, the QoS guaranteed path selection algorithm and the local routing change algorithm, a QoS-guaranteed intelligent routing mechanism in SDN (QI-RM) is first proposed in this paper. The main contributions of this paper are:

- Innovatively combining a variety of simple machine learning methods, a concise data flow classification algorithm, namely MACCA2-RF&RF, is proposed, which can not only improve accuracy but also reduce time consumption when obtaining QoS requirements for data flows.
- The data flow classification algorithm based on machine learning is novelly combined with SDN to achieve a concise and accurate classification of data flows in SDN, based on which, the QoS guaranteed path selection algorithm is proposed to implement QoS-guaranteed routing in IoT.
- Aiming at the situation that the link is congested and QoS cannot be guaranteed, the local routing change algorithm is newly proposed, which only adjusts the links before and after the congested link instead of the entire path, which can reduce the signaling burden of the SDN controller and keep the network stable.

The rest parts of this paper are arranged as follows: Section 2 summarizes the related work. Section 3 describes the problems of data flow classification and QoS guaranteed routing algorithms. Section 4 introduces the QI-RM mechanism proposed in this paper. Section 5 is the evaluation results and analysis of QI-RM. Section 6 is the conclusion of this paper.

2. Related work

For traditional networks, there are some routing algorithms that guarantee QoS. For example, in paper [7], the authors proposed an adaptive QoS-based routing algorithm called AQRV, which adaptively choosed the intersections through which data packets pass to reach the destination, and the selected route should satisfy the QoS constraints and fulfill the best QoS in terms of three metrics, namely connectivity probability, packet delivery ratio and delay. In paper [8], the authors proposed prediction based coding in application layer, opportunistic routing in network layer and Laplacian scaling based on round trip time measurement in session layer. The proposed cross layer solution ensures application specific QoS guaranteed fault tolerance in mobile wireless sensor network.

In terms of SDN routing, load balancing, QoS requirements, link recovery, etc. all need to be considered. The ECMP algorithm can realize multi-channel load balancing through Hash algorithm [9], however, Hash collisions may still be forwarding paths that do not guarantee QoS requirements for data flows. Deng G C proposed a SDN based routing algorithm for IoT using perceived QoS [10]. Park proposed the Network Situation-Aware Framework (NSAF) to more efficiently handle application routing based on the QoS requirements and changing network status, which is used in SDN [11]. Lu qidi used A* algorithm and Yen algorithm to realize path selection and redundant path finding [12], however, the calculated redundant path may not guarantee QoS requirements of data flow in some cases. Swarm intelligence algorithm is also often used for link load balancing algorithm. Wang C proposed a link load balancing algorithm based on ant colony optimization [13]. In this algorithm, the link load, delay and packet loss are taken as the factors affecting the selection of the next node. In paper [14], the PSO algorithm was used to optimize the load balance of network. However, ACO and PSO are easy to fall into local extremum, resulting in poor convergence and low accuracy [15,16]. Wang Likun proposed a collaborative link fault recovery scheme [17]. The scheme proposed that when a link was congested or failed, the link connection should be restored first and the QoS requirement of data on the link should be considered secondly. QoS guaranteed has certain hysteresis. In paper [18], a proactive routing scheme is proposed by correlating the user with the class of the flow that is normally requested, so as to realize a quick-response proactive scheme. However, the proactive scheme needs to consume a large amount of switch storage space.

The basis of QoS guarantee of data flow is to accurately identify QoS requirements of data flow. By classifying the network data flow, the application category of the data flow can be identified and the QoS requirements can be obtained. At present, researchers have proposed a lot of data flow classification algorithms. Li G used Deep Packet Inspection (DPI) to classify data flows in SDN [19]. This traffic classification method based on DPI mainly detected the load of the application layer to distinguish the program from the service, but the classification process consumed a lot of computing resources. Amaral P used SDN controller to collect data flows and extract features, and finally classified the flows by using random forest algorithm, Stochastic Gradient Boosting and Extreme Gradient Boosting to classify flows respectively [20]. As can be seen from their results, the accuracy of these three methods is not very high.

With the development of machine learning (ML) and deep learning, they are applied to the data flow classification algorithms. For example, in paper [21], the authors identified the issues in ML-based traffic classification (TC) in order to devised the best solution; i.e. the TC framework should be scalable to accommodate network expansion, can accurately identify flows according to their source applications, while maintaining an efficient run-time and memory requirement. In paper [22], the FlowSeer is proposed, which is a fast, low-overhead flow detection and scheduling system using data stream mining. The features from flows' first few packets allow is used to train the data flow classification models that can accurately and quickly predict the rate and duration of any initiated flow. Wang P proposed to classify network applications in SDN by deep learning [23], which required no artificial feature selection and extraction, and had high classification accuracy. However, it is well known that parameter optimization and model training of deep learning are very time-consuming. José Suárez-Varela proposed a method combining DPI with machine learning [24]. But this approach struck a balance between precision and performance, it required support from more modules, such as the DNS module, which added to the complexity of the system. Lin S C proposed a semi-supervised machine learning traffic classification framework based on QoS perception [25], but the framework's accuracy was less than 95%. However, in the above paper, authors did not consider that in order to realize the routing mechanism based on data flow classification in SDN, the SDN controller should satisfy both high accuracy and low time consumption for data flow classification.

Therefore, how to efficiently get the QoS requirements of data flows and realize the SDN scheme to guarantee that QoS requirements have become the research focus of this paper.

3. Problem description

In SDN, the controller, like the brain of SDN, obtains information of data flows and link status in the network, and controls the switches for data forwarding [22]. The controller is programmable, so routing algorithms can be designed according to requirements. The routing algorithm in SDN can select the path according to the QoS requirements of data flows. However, different data flows have different QoS requirements, how to obtain the QoS requirements of the data flows is one of the problems to be solved by the QoS guaranteed mechanism in SDN. How to route the data flows with different QoS requirements in order to guarantee the QoS is another problem that the QoS guaranteed mechanism in SDN needs to solve. When the link is congested, how to adjust the route in time to guarantee QoS as much as possible is another problem to be solved.

3.1. Problems of data flow classification methods

QoS requirements are determined by the categories of data flows, and the categories can be obtained by the data flow classification methods. Some traditional data flow classification methods (such as depth package inspection) have problems of violating user privacy and taking a long time for classification. With the popularization of machine learning (ML) technology, statistical and behavioral models can be constructed accurately and efficiently. Data flow classification technology based on ML becomes an effective method [26].

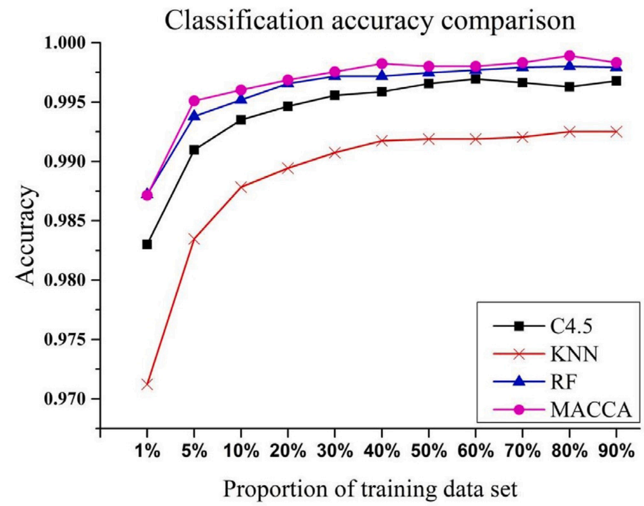


Fig. 1. Classification accuracy comparison of MACCA, C4.5, KNN and RF on Moore dataset.

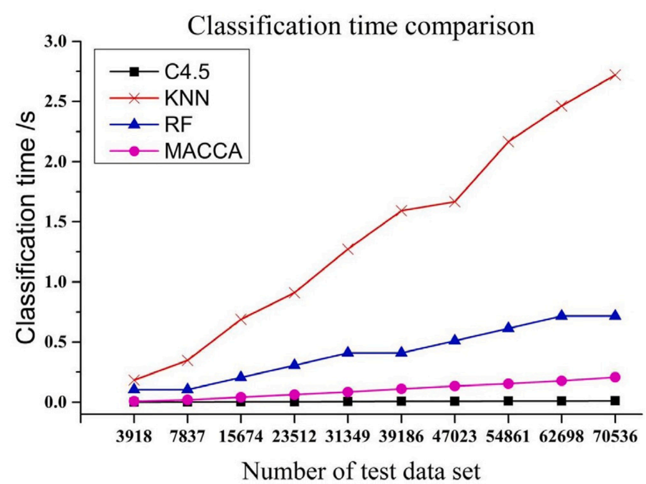


Fig. 2. Classification time comparison of MACCA, C4.5, KNN and RF on Moore dataset.

Classification algorithms based on ML need to ensure high accuracy and low time consumption in SDN scenario. Aiming at this problem, C4.5, Naïve Bayesian, KNN and random forest algorithm were compared on Moore dataset in our previous papers [6], and we concluded in our previous paper that it was difficult to achieve high accuracy and low consumption at the same time for a single machine learning algorithm. Therefore, the misclassification aware collaborative classification algorithm (MACCA) was proposed and used to identify the categories of data flows. The performance comparison results of MACCA algorithm are shown in Figs. 1 and 2.

MACCA collects complete data of the data flow for feature extraction, which can improve the accuracy. However, it requires a long data collection time, which is not conducive to the realization of QoS guaranteed for data flows.

In order to reduce the data collection time, a suitable data flow classification algorithm should collect a small number of data in the data flows. The Li dataset used the 12 features extracted from the first 5 packets of the data flow to classify the data flow on the basis of Moore [27]. As shown in Fig. 3, in the case of using Li dataset to evaluate the MACCA, which means a small number of data in the data flows is selected by MACCA to classify the data flows, the accuracy of MACCA is lower than that of RF algorithm, which is an important part of MACCA.

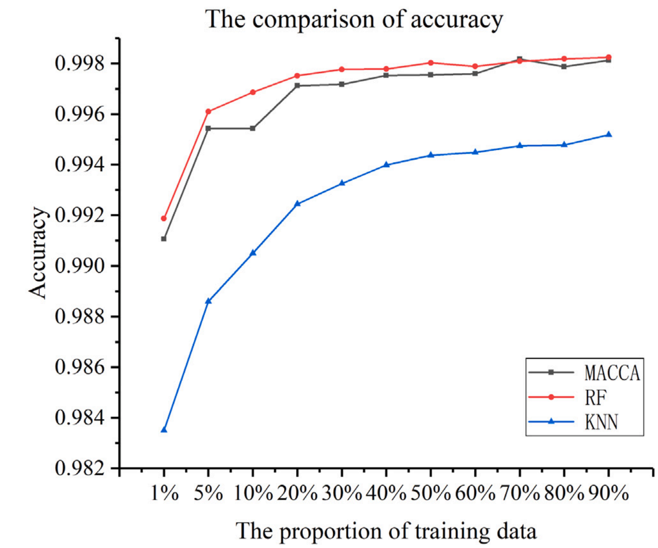


Fig. 3. Classification accuracy comparison of MACCA, KNN and RF on Li dataset.

Therefore, MACCA's advantage is not obvious in the case of selecting a small number of data in data flows. MACCA should be trained with datasets with few data features and further improved.

3.2. Problems of QoS guaranteed routing algorithms

A suitable QoS guaranteed routing mechanism in SDN needs to select the forwarding path for different data flows, considering the network link state and QoS requirements. At present, the most commonly used routing algorithms in SDN are Dijkstra algorithm and ECMP algorithm [28]. However, the traditional Dijkstra algorithm only considers a single factor in the path selection. Dijkstra algorithm used in most SDN controllers takes hops as the weight, which only considers hops but fails to consider the QoS requirements of the data flow and network link state. Therefore, the traditional Dijkstra algorithm selects a path that cannot guarantee QoS requirements. Similarly, the ECMP algorithm selects the forwarding path for the data flow through the Hash algorithm, but it still fails to take into account the data flow QoS requirements and network link state.

In order to implement the routing algorithm to guarantee QoS requirements, the routing algorithm should consider the QoS requirements (bandwidth, delay, and packet loss rate in this article) in the path as the weight. Therefore, a new metric $L^{std}_{(i,j)} = (b^{std}_{(i,j)}, d^{std}_{(i,j)}, l^{std}_{(i,j)})$ should be developed, where $b^{std}_{(i,j)}$, $d^{std}_{(i,j)}$ and $l^{std}_{(i,j)}$ represent bandwidth, delay, and packet loss rate between node i and node j on the link respectively, and $L^{std}_{(i,j)}$ represent the weight of the link. In addition, the load of switches i should also be considered, which can be represented by the total bandwidth of the normalized flow table $FwdRate^{std}_i$. So, the total cost of the path $Cost_{total}$ can be defined by (1), where i and j are nodes on the path. The smaller $Cost_{total}$ is, the more likely the path is to guarantee the QoS requirements.

$$Cost_{total} = \sum_{(i,j)} L^{std}_{(i,j)} + \sum_{i} FwdRate^{std}_i \tag{1}$$

Therefore, the first problem the QoS guaranteed routing algorithm should solve is that the QoS guaranteed routing algorithm should consider the comprehensive metric $Cost_{total}$, and make the $Cost_{total}$ of the selected path as small as possible.

When the links are congested, a suitable QoS guaranteed routing mechanism also needs to be able to change the forwarding path of the data flow on the congestion links in order to guarantee the QoS requirements. Most of the existing algorithms only avoid link congestion in the path selection, and lack the ability of data forwarding and adjustment after link congestion.

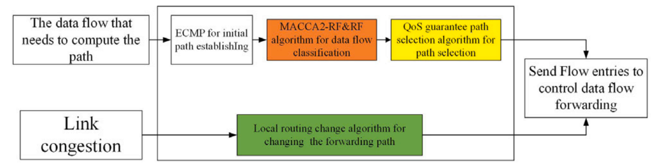


Fig. 4. The working process of QI-RM.

However, if the data flow path is changed, the larger the change range is, the greater the impact on the network will be, and the more control signaling is needed, the more difficult it will be to manage the whole network. At the same time, the signaling pressure of the controller and flow table management problems should also be considered. The more switches carry out forwarding adjustment, the more uncertainty exists in the network, and the greater the signaling pressure of the controller is. So, when the controller changes the forwarding path of data flows, it should use less flow table items (denoted as $FlowEntries_{total}$) to control the forwarding of data flow.

Therefore, the second problem the QoS guaranteed routing algorithm should solve is that when the link is congested, the routing algorithm should change the forwarding path as little as possible and make $FlowEntries_{total}$ as small as possible.

Based on the analysis of the above, the overall goal of a QoS guaranteed routing algorithm is defined as P1, and the constraints are C1 to C7:

(P1) $min(\omega_1 * Cost_{total} + \omega_2 * FlowEntries_{total})$

(1) The weight of the link $L^{std}_{(i,i+1)} = (b^{std}_{(i,i+1)}, d^{std}_{(i,i+1)}, l^{std}_{(i,i+1)})$ in the forwarding path can meet the QoS requirements of data flow.

(2) $bw_{path} > bw^{threshold}_n$.

(3) $delay_{path} < delay^{threshold}_n$.

(4) $loss_{path} < loss^{threshold}_n$.

(5) $\omega_1 + \omega_2 = 1, 0 < \omega_1, \omega_2 < 1$

P1 means that the overall goal of QoS guaranteed routing is to minimize the weighted sum of $Cost_{total}$ and $Flowentries_{total}$, where ω_1 and ω_2 are weights, so that the network can guarantee the data flow QoS requirements and reduce the signaling burden of the SDN controller before and after the congestion. Constraint (1) requires that the remaining bandwidth, delay, and packet loss rate of $L^{std}_{(i,i+1)}$ all can meet the QoS requirement threshold of data flow. Constraint (2) to (4) constrain the bottleneck bandwidth, total delay and expected packet loss rate (denoted as bw_{path}, $loss_{path}$ and $loss_{path}$) of the selected path to meet the thresholds (denoted as $bw^{threshold}_n$, $loss^{threshold}_n$ and $loss^{threshold}_n$). Constraint (5) requires that the sum of ω_1 and ω_2 is equal to 1. Both ω_1 and ω_2 are greater than 0 and less than 1. There is no strict restriction on $FlowEntries_{total}$. $FlowEntries_{total}$ should be as few as possible.

4. Design of the QI-RM

The QI-RM is mainly composed of MACCA2-RF&RF algorithm, QoS guarantees path selection algorithm and local routing adjustment algorithm. The working process of the QI-RM is shown in Fig. 4. When a new data flow comes, the ECMP algorithm is first used to select the initial path for data flows. Then, during forwarding, the controller samples the data to obtain the relevant features. Then, the improved data flow classification method (named MACCA2-RF&RF) is used to classify the data flow and get its QoS requirements and the threshold. If the current initial path can meet its QoS requirements, the path will not be changed. Otherwise, the QoS guaranteed path selection algorithm is used to select the forwarding path for the data flow, and the corresponding flow entries will be sent to switches to forward the

Table 1
12 features in Li dataset.

Features	Feature description
server Port	Destination port
client Port	Source port
actual_data_pkts_a_b	Number of packets with at least one byte of TCP segment in the payload(source-to-destination)
pushed_data_pkts_a_b	Number of packets with TCP header PSH bit (source to destination)
pushed_data_pkts_b_a	Number of packets with TCP header PSH bit (destination to source)
min_segm_size_a_b	Minimum segment observed (source to destination)
avg_segm_size_b_a	Average segment observed (destination to source)
initial_windows-bytes_a_b	Total number of bytes sent in the initial window (source to destination)
initial_windows-bytes_b_a	Total number of bytes sent in the initial window (destination to source)
RTT_sample_a_b	Total number of RTT samples (source to destination)
med_data_ip_a_b	The median of the total number of bytes in the payload of the IP datagram (source to destination)
var_data_wire_b_a	Ethernet frame's byte variance (destination to source)

data flow. If the controller detects the congestion of network link, the local routing change algorithm is used to select the data flow and calculate the transfer path, and the flow entries will be sent to switches to forward the data flow.

In this section, the MACCA2-RF&RF algorithm, QoS guaranteed path selection algorithm and local routing change algorithm will be described in detail, respectively.

4.1. MACCA2-RF&RF data flow classification algorithm

In paper [6], through feature selection, 14 of the 248 features were selected from the Moore dataset for data flow classification, and the MACCA was proposed. However, the Moore dataset is a dataset with 248 features obtained by collecting the data of a complete TCP data flow. In order to realize the QoS guaranteed routing for the data flow, it is obviously inappropriate for the controller to collect all packets of the data flow and classify the data flow. The Li dataset used the 12 features extracted from the first 5 packets of the data flow to classify the data flow on the basis of Moore. The Li dataset is closer to the scenario of QoS guarantee of SDN routing. The 12 features in the dataset are shown in Table 1.

The proposed MACCA consists of base classifier module, misclassification results judgment module and decision module. The base classifier module is composed of C4.5 and Naïve Bayes (NB) algorithm, misclassification results judgment module is composed of C4.5 algorithm, and the decision module is composed of RF algorithm. In MACCA, the base classifier classify the data initially. The misclassification results judgment module judges the accuracy of the classification results of the two base classifiers. Based on the judgment results, the decision module chooses to output the correct result or re-classify it. The details of MACCA are introduced in our previous paper [6].

In order to improve the shortcomings of MACCA introduced in Section 3, the modules in MACCA are modified to design four data flow classification algorithms originating from MACCA, which are named MACCA2, MACCA2-CART &RF, MACCA2-RF&RF and MACCA3 respectively. The relationship between these four algorithms and MACCA is shown in Fig. 5.

In Fig. 5, MACCA2 is obtained by MACCA replacing the C4.5 algorithm in the base classifier module with the CART algorithm. MACCA2-CART&RF is obtained by MACCA2 replacing the C4.5 algorithm in the misclassification results judgment module with the CART algorithm. MACCA2-RF&RF is obtained by MACCA2 replacing the C4.5 algorithm in the misclassification results judgment module with the RF algorithm. MACCA3 is obtained by MACCA replacing the NB algorithm in the base classifier module with the CART algorithm and the C4.5

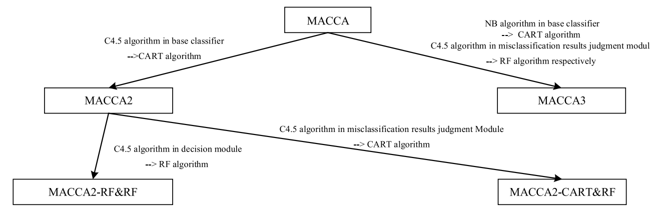


Fig. 5. The relationship between MACCA and MACCA2, MACCA2-CART&RF, MACCA2-RF&RF, MACCA3.

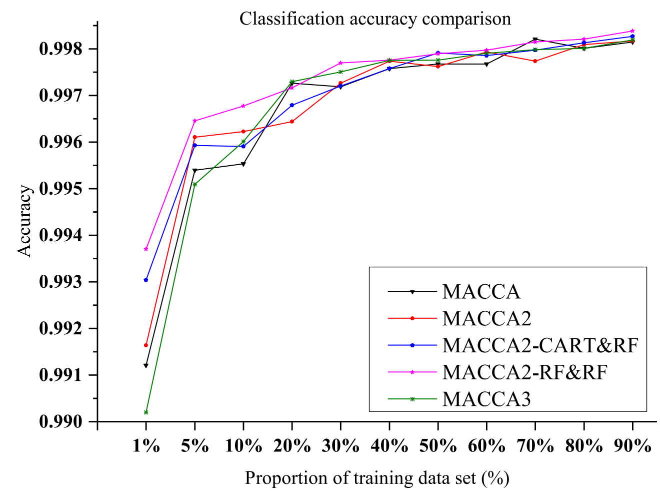


Fig. 6. Classification accuracy of MACCA, MACCA2, MACCA2-CART&RF, MACCA2-RF&RF and MACCA3 on Li dataset.

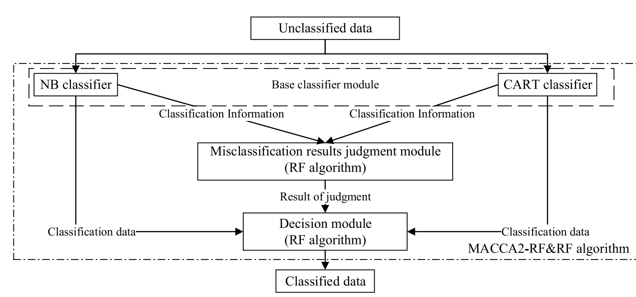


Fig. 7. The working process of MACCA2-RF&RF.

algorithm in the misclassification results judgment module with the CART and RF algorithms.

We compared the accuracy of MACCA2, MACCA2-CART&RF, MACCA2-RF&RF, MACCA3 and MACCA on the Li dataset. The results are shown in Fig. 6. It can be seen that the accuracy of MACCA2-RF&RF algorithm remains the highest among these algorithms. And in the case of less data, the MACCA2-RF&RF has a greater advantage. The average accuracy of MACCA is 99.67%, and the average accuracy of RF algorithm is 99.71%.

Therefore, the proposed MACCA-RF&RF algorithm is used to obtain the QoS requirements through data flow classification in the proposed QI-RM. The working process of MACCA2-RF&RF is shown in Fig. 7.

The pseudocode of MACCA2-RF&RF is shown in Table 2.

Lines 1 to 3 in Table 2 indicate that for unclassified data, use two basic classifiers to classify, and use the category labels Label1 and Label2. The base classifier module generates classification information data and classification data. Line 4 in Table 2 indicates that the misclassification result judgment module obtains the judgment result based on the classification information data of the base classifier. Lines 5 to

Table 2
The pseudo-code of MACCA2-RF&RF.

MACCA2-RF&RF Algorithm
Input: A Data with 14 features
Output: Class label
1.Label1←CART classifier classification result.
2.Lable2←Naïve Bayesian classifier classification result.
3.Base classifier module generates classification information and classification data.
4.Result←MisclassificationResultsJudgmentModule (classification information)
5.If Result ==0 then
6. Final_result←Decision module recalculates classification result.
7.Else if Result ==1 or Result ==3 then
8. Final_result←Label1.
9.Else if Result ==2 then
10. Final_result←Lable2.
11.Return Final_result.

14 in Table 2 indicate that the decision module decided to recalculate the final classification result, or directly output the final result based on the judgment result.

In the QI-RM, the QoS requirements of different service data flows are classified in order to refine QoS assurance operations for different data flows. And different link parameter weight vectors can be set for each QoS requirements (bandwidth, delay, packet loss rate in this article) in order to meet the link performance requirements of data flows with different QoS requirements, which is conducive to the implementation of QoS guaranteed path selection algorithm. In this paper, QoS requirements threshold value is also set for each class of QoS requirements in order to provide guarantee of minimum QoS requirements for data flows, so as to guarantee the normal operation of IoT services. When classifying data flows, if the scope of a class is too large, it may cause too many data flows of different categories to belong to the same class, which is not conducive to guarantee QoS. This paper refers to the data flow classification types of 3GPP and classifies the data flow types, mainly including four categories: session services, streaming media services, interactive services and background services. Among them, session services include SERVICES, CHAT, VOIP and other applications; streaming media services include MULTIMEDIA applications; interactive services include WWW, DATABASE, P2P, INTERACTIVE, GAMES and other applications; background services include BULK, MAIL and other applications. QoS requirements for different services are different. Session services have a larger requirement for delay, while the requirements for bandwidth and packet loss rate are not large. Streaming media services also has certain requirements for delay, but it is lower than the session services, and has no high requirement for packet loss rate. Interactive services have a large requirement for bandwidth, as well as the requirements for delay and packet loss rate. Background services have a high requirement for packet loss rate, but low requirements for bandwidth and delay.

Through further analysis, this paper sets 11 classes of QoS requirements classes according to the data flow categories on the basis of the above four services. Among them, QoS 1 to 3 and 8 correspond to interactive services, QoS 4 correspond to streaming media services, QoS 5 to 7 correspond to session services, and QoS 9 and 10 correspond to background services. In this paper, QoS 11 is set as dropped packets, mainly because this class corresponds to malicious attacks on network data flow, which is dropped to ensure network security. For example, considering that UDP data flow has a large requirement for delay, this paper classifies UDP data flow into QoS 8. This paper sets different thresholds for each kind of QoS requirement classes, as shown in Table 3. Setting threshold can ensure the minimum QoS requirements of network data flows and the normal operation of relevant services. At the same time, before the calculation of the forwarding path, the link can be filtered according to the threshold to improve the efficiency of path search. In this paper, 11 classes of QoS requirements classes are set according to data flow categories on the basis of the above four services, which can further refine QoS guarantee operation and realize a better QoS guarantee service. What is more, the 11 classes of QoS can also meet the needs for data transmission in IoT.

Table 3
QoS requirements threshold for different QoS requirements classes.

QoS class	Minimum bandwidth (kbps)	Maximum delay (ms)	Maximum packet loss rate (%)
QoS 1	200	300	3
QoS 2	150	200	3
QoS 3	150	200	2.5
QoS 4	128	150	2.5
QoS 5	100	150	2.5
QoS 6	100	110	2.5
QoS 7	100	150	3
QoS 8	80	110	1.5
QoS 9	50	150	2
QoS 10	50	200	1
QoS 11	Drop	Drop	Drop

After the data flow categories and QoS requirements are identified, the QoS guaranteed path selection algorithm in the QI-RM select the forwarding path according to the QoS requirements and threshold of the data flow.

4.2. QoS guaranteed path selection algorithm

For the defects of Dijkstra algorithm and ECMP algorithm in the problem description in Section 3, in order to implement the routing algorithm to guarantee QoS requirements, considering the QoS requirements of different data flows for link bandwidth, delay and packet loss rate, for data flows, the link after normalization parameters $L_{(i,j)}^{std}$ weight can be represented as (2).

$$Score_{(i,j)} = L_{(i,j)}^{std} \cdot W_i \tag{2}$$

Among them, the vector $W_i = (w_{i1}, w_{i2}, w_{i3})^T$ represents the link evaluation weight vector of QoS i data flow, w_{i1}, w_{i2}, w_{i3} respectively represent the weight vector of link residual bandwidth, delay and packet loss rate under the QoS requirement, while $0 < w_{i1}, w_{i2}, w_{i3} < 1$ and $w_{i1} + w_{i2} + w_{i3} = 1$.

The load of the switch feedback the path load. The load of switch i can be represented by the normalized total bandwidth of the flow table $FwdRate_i^{std}$. $FwdRate_i^{std}$ can be calculated by (3)–(5).

$$BW_{(i,j)}^{total} = \sum_{m=1}^{N} \frac{byte\_count_i^m * 8}{duration\_sec_i^m + duration\_nsec_i^m * 10^{-9}} \tag{3}$$

$$FwdRate_i = BW_{(i,j)}^{total} * 10^{-3} \tag{4}$$

$$FwdRate_i^{std} = \frac{FwdRate_i}{FwdRate_{max}} \tag{5}$$

Among them, N is the total number of flow table items in switch i, $byte\_count_i^m$ represents the number of bytes that have been matched by data flow m in switch i, $duration\_sec_i^m$ and $duration\_nsec_i^m$ respectively represent the time that the corresponding flow table item has existed, and the units are seconds and nanoseconds, respectively. $BW_{(i,j)}^{total}$ is the sum of the width of the flow meter of switch i, $FwdRate_i$ represents the total bandwidth of the flow table of switch before normalization, which is reduced by an order of magnitude by $BW_{(i,j)}^{total}$. $FwdRate_{max}$ is the largest bandwidth of any switch in the network.

So, the sive weight of the path from the source switch to any switch i is as shown in (6).

$$CW_{src,i} = Score_{(src,i)} + s * FwdRate_i^{std} \tag{6}$$

Among them, s is the constant coefficient, which represents the weight of the link load when evaluating the link state. Through many

Table 4
The pseudo-code of QoS guaranteed path selection algorithm.

QoS guaranteed path selection algorithm
Input: Src, Dst, QoS_i, G
Output: newPath
1.newPath←null
2.If the path from Src to Dst is existed in the path list then:
3. newPath ← the path from Src to Dst
4.End if
5.If newPath doesn't meets the requirements or newPath is null:
6. G'←G is processed according to the QoS_i,and the bandwidth utilization of $L_{(i,j)}$
7. index← −1
8. Q←Init-Queue (G',Src).
9. While Q is not null then:
10. index← EXTRACT-MIN(Q)
11. For each neighbor $switch_m$ of $switch_{index}$ in G':
12. $CW_{(src,index)} + CW_{(index,m)} < CW_{(src,m)}$ then:
13. $CW_{(src,m)} \leftarrow CW_{(src,index)} + CW_{(index,m)}$
14. DECREASE-KEY(Q, m, $CW_{(src,m)}$)
15. Path.add(m)
16. End if
17. End for
18. End while
19. Update path list according to Path.
20. newPath←the path from Src to Dst in Path
21.End if
22.Return newPath

experiments, s is roughly determined to be 0.1. In different specific scenarios, the value of s may be different. In this paper, the adaptive adjustment of the optimal parameters is not considered.

The QoS guaranteed path selection algorithm uses the above comprehensive weight to choose the path. Before searching a path, the algorithm searching the path from the active switch to the destination in the existing path list. If there is a complete path, it will check whether the existing path can guarantee the QoS requirement. If it does, the algorithm is terminated and the path is returned. Otherwise, graph G will be processed according to the QoS requirement threshold of the data flow and the bandwidth utilization of the link, and the comprehensive weight of the link that does not meet the QoS requirement threshold will be set to 10 000, so as to ensure that the path in the final result can guarantee the QoS requirement. Then, Dijkstra algorithm based on comprehensive weight is used to search the path. Finally, return the forwarding path and update the list of paths based on the search results.

In order to reduce the search time of Dijkstra algorithm, a small root heap is used to store unselected switches. Since Fibonacci heap only needs $O(1)$ amortized time for operations that do not involve deleting elements, it only has $O(log_2 N)$ amortized time for operations such as deletion. Therefore, in this paper, a Dijkstra algorithm based on Fibonacci heap is used to search the path. The pseudocode of the QoS guaranteed path selection algorithm is shown in Table 4.

Lines 1 to 5 in Table 4, the algorithm checks whether there is a path in the path list and detects whether the path meets the requirement. Line 6 in Table 4 indicates that before the path search, the existing network topology graph G is processed to obtain a new network topology graph G' composed of links that can meet the data flow QoS requirements. Lines 7 to 18 in Table 4, the algorithm enters the path search stage, and the comprehensive weight is calculated iteratively. Lines 19 to 22 in Table 4 indicate updating the path list, getting the forward path newPath, and returning the forward path newPath.

After getting the forward path, controller can send the flow entries to change the forwarding path of the data flow. When the network link is congested, the local routing change algorithm is needed to change the path of data flows on the congestion link.

4.3. Local routing change algorithm

In order to better identify network link congestion, the choice of congestion threshold is very important. When link bandwidth utilization is above the threshold, new data flows are no longer forwarded through the link. The congestion threshold limits the maximum available bandwidth of the link. The larger the threshold, the higher the available bandwidth of the link and the greater the overall throughput of the network will be. However, the larger the threshold, the smaller difference between the congestion threshold bandwidth and nominal bandwidth in the case of congestion. When the actual bandwidth utilization exceeds the threshold, the greater the possibility of a large packet loss rate, and the smaller the cache time left for routing change. Therefore, in this paper, the congestion threshold is set at 90%.

When the controller detects the link congestion, the controller can use local routing change algorithm many times to change the data flow path in order to reduce congestion and guarantee the QoS demand of data flow. In addition to the controller congestion detection process, the process of local routing change algorithm is as follows:

Firstly, the traffic information on the congestion link is obtained and the data flow selection is carried out. After the link congestion, it is common to change the forwarding path from the bandwidth-intensive data flow. However, for data flow with large bandwidth, there may not be a better link in the network to guarantee its QoS requirements, and controller needs to carry out flow selection many times and it will increase the time consumption. Starting from a data flow with a small bandwidth will speed up the routing process. For some data flows with small bandwidth and delay or packet loss rate sensitivity, QoS guarantees after congestion can be realized more quickly. Therefore, in the QI-RM, the data flow with small bandwidth is selected first to change the forwarding path. If the bandwidth difference between the data flows is small, the QI-RM will choose one according to the QoS class of the data flows.

Then, after selecting the data flow that needs to be adjusted, the transfer path needs to be calculated. If the forwarding path is calculated for the data flow from the source switch to the destination switch again, too many switches need to modify the flow table, which is not conducive to network management. However, local route change is carried out only around the congestion link, which is easy to manage. In the local routing change algorithm, for the congested link (i, j), the sizes of $FwdRate_i^{std}$ and $FwdRate_j^{std}$ of the two switches are compared, and the larger one is considered to have heavier load. The local routing change algorithm requires recalculating the path for the selected data flow to bypass the relatively heavily loaded switches. There are four situations when calculating the transfer path:

(1) Search forward

The first case is shown in case 1 in Fig. 8. If switch i is in a state of high load, then our algorithm selects the previous hop switch m of switch i as the source and switch j as the destination, and uses QoS guaranteed path selection algorithm to calculate the transfer path. If a hop forward does not find a transfer path that satisfies QoS well, the search continues until the entry switch.

(2) Search backward

The second case is shown in case 2 in Fig. 8. If switch j is under high load, our algorithm takes switches i as the source, selects the next hop switch n of switch j as the destination, and uses QoS guaranteed path selection algorithm to calculate the transfer path. If a hop back does not find a transfer path that satisfies QoS well, the search continues until the exit switch.

(3) Search forward and backward

The third case is shown in case 3 in Fig. 8. If switch i and switch j are under high load, our algorithm selects switch m of the previous hop of switch i as the source and switch n of the next hop of switch j as the destination, and uses QoS guaranteed path selection algorithm to calculate the transfer path. If it does not, then it goes to case 4.

(4) N hops forward or backward

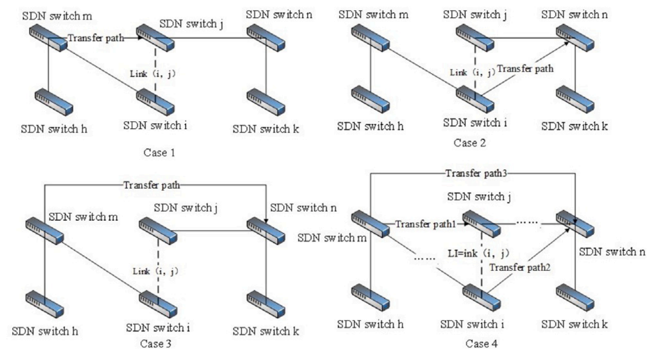


Fig. 8. Four cases of the transfer path.

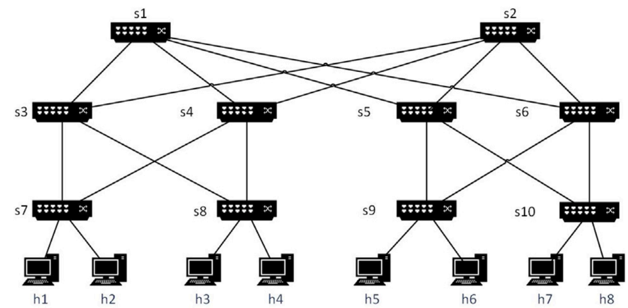


Fig. 9. Simulation experiment network topology.

Table 5
The pseudo-code of local routing change algorithm.

Local routing change algorithm
Input: G, oldPath, $L_{(i,j)}$
Output: newPath
1.newPath ← null;
2.FlowInfo ← Get all flow information on $L_{(i,j)}$;
3.ChosenFlow ← Choose flow according to FlowInfo;
4.ChosenNode ← Choose $switch_j$ according to $FwdRate_j$;
5.Src ← Choose TransferPath src according to ChosenNode;
6.Dst ← Choose TransferPath src according to ChosenNode;
7.While newPath doesn't meets the requirements or newPath is null:
8. TransferPath ← Calculation TransferPath using QoS guaranteed path selection algorithm (mentioned in Section 4.2) according to Src, Dst, ChosenFlow's QoS, G;
9. newPath ← Replace congestion path in oldPath with TransferPath;
10. If newPath doesn't meets the requirements then:
11. Chang Src or Dst according to ChosenNode;
12. End if
13.End while
14.Return newPath;

The fourth case is shown in case 4 in Fig. 8. If switches i and j are under a state of heavy load, our algorithm chooses switch i jumped on the switch as the source, but if the hop and the next-hop switch is also in a high load, the forward and backward search n hops switches at the same time, uses the QoS guaranteed path selection algorithm to calculate the transfer path 1, 2, 3, choose path which has smaller comprehensive weight as the final transfer path.

Finally, update the old forwarding path with the transfer path to get the newPath. When newPath can secure QoS requirements, the algorithm terminates and returns the newPath. If newPath cannot guarantee QoS requirements, set the comprehensive weight of the link with the largest comprehensive weight in newPath to 10 000, and then change the source or destination of the transfer path according to the selected switch and repeat the above process.

The pseudocode of the local routing change algorithm is shown in Table 5.

Line 1 in Table 5 indicates that the adjusted path newPath is set to null. Line 2 in Table 5 indicates that the algorithm obtains the data flow information on the congestion link. Line 3 in Table 5 is to select the data flow. Lines 4 to 6 in Table 5 indicate that the algorithm selects switch and the source and destination switches of the transfer path. Lines 7 to 13 in Table 5 represent the calculation of the transfer path. After the transfer path is obtained, the old forwarding path oldPath of the data flow is updated according to the transfer path, and the adjusted path newPath is obtained. Finally, return the new forward path newPath.

After the controller gets newPath, it can send the flow entries to control the data flow forwarding. When the link is congested, the controller may use the algorithm many times until the link is non-congested. And in the process of multiple adjustments, it is not necessary to collect FlowInfo and choose a switch for many times.

5. Evaluation

In this Section, the simulation scenarios and parameters of this paper are first introduced in Section 5.1. The environment is Windows 10 OS, CPU i7 9750h, 32 GB memory. The results and analysis of the experiment are introduced in Section 5.2.

5.1. Scenarios and parameters

The experiments and simulation for the MACCA2-RF&RF and the QI-RM are carried out. In the experiments of data flow classification algorithm, 508,385 pieces of data were collated by combining the data of Day 2, Day 3 and Site B in Li dataset [27], and tested the classification performance of CART, KNN, Random Forest, CART-naive Bayes-random Forest voting classification algorithm, MACCA2-CART&RF and MACCA2-RF&RF in this paper. MACCA2-CART&RF represents the algorithm of replacing judgment module C4.5 algorithm with CART algorithm. MACCA2-RF&RF is the algorithm used in this paper, while MACCA2-CART&RF is one of the contrast algorithms. In the experiment, the Python sklearn library was used to implement all the algorithms. In the experiment of classification time, the training set and the testing set were randomly selected. When evaluating classification time, in order to get closer to the real IIoT scenario, the training set is relatively large, and the testing set is relatively small. Because in the actual IoT scenario, the data flows are continuous and the amount of data is very large. The proportion of the test set in the dataset was 0.6 and that of the training set is 0.4. 10 groups of experiments were set up according to the proportion of datasets other than the test set and the training set, including 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9. The number of test dataset is between $508\,383 * 0.6 * 0.05 = 15\,252$ and $508\,383 * 0.6 * 0.9 = 274\,528$.

In the simulations of QI-RM, Mininet was used to set up the network topology that is shown in Fig. 9, which contains 10 switches, each boundary switches connected to the two hosts, link bandwidth, switches between the client and the same link set, bandwidth at 100 Mbps, without delay, packet loss. Link parameter settings between switches are shown in Table 6. The controller uses a Floodlight controller. According to Table 3, QoS 1 data flow, QoS 6 data flow, and QoS 10 data flow have different requirements on bandwidth, delay, and packet loss rate, respectively. Therefore, QoS 1 data flow, QoS 6 data flow and QoS 10 data flow were used to verify the performance of the mechanism.

5.2. Experiment results and analysis

In this section, in order to more clearly evaluate the accuracy and classification time of the proposed data flow classification algorithm, MACCA2-RF&RF is simulated separately in Section 5.2.1. And then, in order to evaluate the overall performance of the proposed QoS guaranteed intelligent routing mechanism, QI-RM is simulated in Section 5.2.2.

Table 6
Link parameter setting of simulation experiment.

Links	Bandwidth (Kbps)	Delay (ms)	Packet loss rate (%)
s1, s3	1	10	1
s1, s4	5	20	0.01
s1, s5	1	10	0.01
s1, s6	1	10	0.5
s2, s3	1	50	1
s2, s4	1	10	0.1
s2, s5	1	10	0.5
s2, s6	5	30	0.1
s3, s7	0.15	20	1
s3, s8	5	20	0.5
s4, s7	10	20	0.01
s4, s8	1	10	1
s5, s9	5	10	0.01
s5, s10	3	10	0.01
s6, s9	1	30	1
s6, s10	5	20	0.5

Table 7
Classification performance comparison on Moore dataset.

Classification algorithm	Average classification accuracy	Average classification time
RF	99.6	0.41
MACCA	99.66	0.11
MACCA2-RF	99.7	0.26

5.2.1. Experiments on the MACCA2-RF&RF

In the experiments, the mentioned average value represents the average value of 11 experimental results. According to the accuracy results in Fig. 10, the average accuracy rates of MACCA2-RF&RF and MACCA2-CART&RF are 99.73% and 99.696% respectively, the average accuracy of random forest and voting classification algorithms are 99.71% and 99.693%, MACCA, CART and The average accuracy of KNN are 99.67%, 99.66% and 99.239%. MACCA2-RF&RF has the highest average accuracy. According to the results of classification time in Fig. 11, the average time consumption of KNN is the highest, the highest time consumption can reach 28.53 s, the time consumption of the voting classification algorithm is second, the classification time of the random forest algorithm is the third highest, and the classification time by voting method the average time for classification higher than the random forest algorithm is 1.06 s. MACCA2-RF&RF and MACCA2-CART&RF classification time are lower than Random Forest's average classification time, with an average difference of 0.30 s and 0.89 s, ranking third and fourth. The average classification time of MACCA algorithm and CART is lower.

In addition, the performance of MACCA2-RF&RF on the Moore dataset is verified. The data in the Moore dataset is obtained by collecting complete data flows and has 14 features. The experimental parameter settings are similar to the above experiments. The experimental results are shown in Table 7. Due to space limitations, this article cannot fully show all the experimental results of MACCA2-RF&RF on the Moore dataset. The average accuracy of MACCA2-RF&RF in the Moore dataset is 99.70%, which is higher than that of MACCA and RF algorithms. And the average classification time of MACCA2-RF&RF is 0.26 s, which is much lower than that of RF algorithms and slightly larger than that of MACCA algorithm. Therefore, on the Moore dataset, MACCA2-RF&RF also has better performance.

According to the above results, whether it is on the Moore dataset or the Li dataset, MACCA2-RF&RF has higher accuracy and lower time consumption than RF algorithm and has higher accuracy then MACCA algorithm. Therefore, for different datasets, MACCA2-RF&RF is effective. MACCA2-RF&RF has better robustness and is more suitable for the QI-RM.

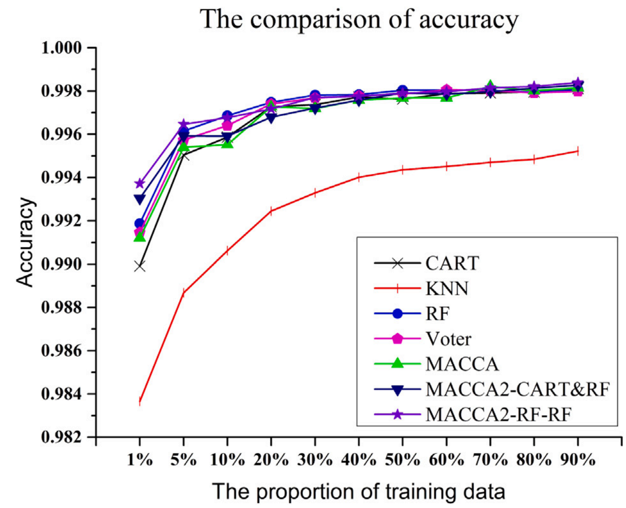


Fig. 10. Classification accuracy comparison.

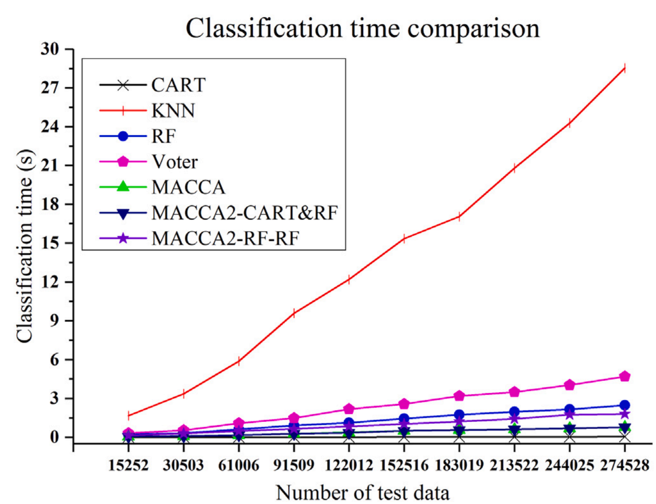


Fig. 11. Classification time comparison.

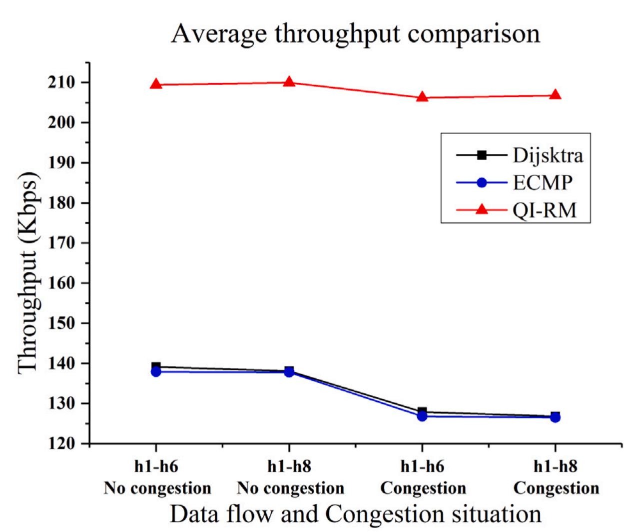


Fig. 12. Average throughput comparison.

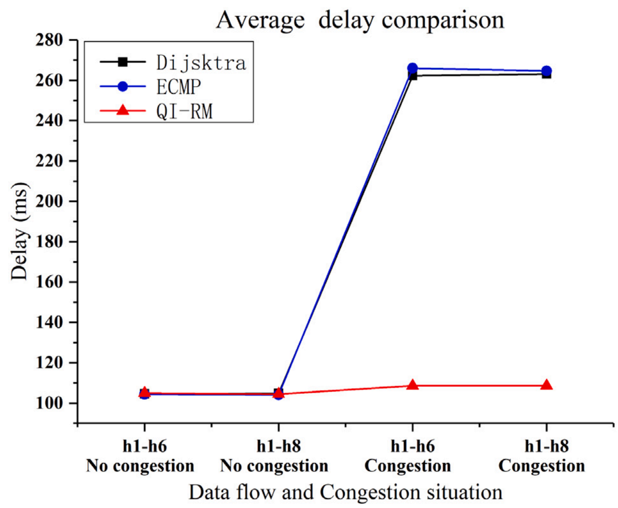


Fig. 13. Average delay comparison.

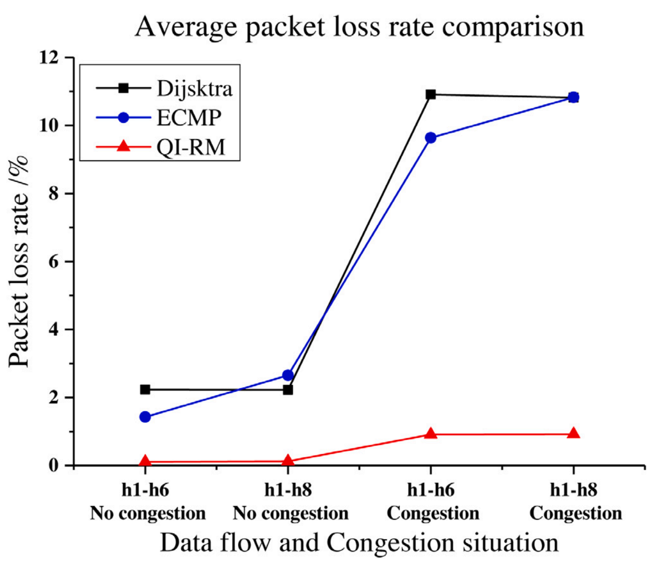


Fig. 14. Average packet loss rate comparison.

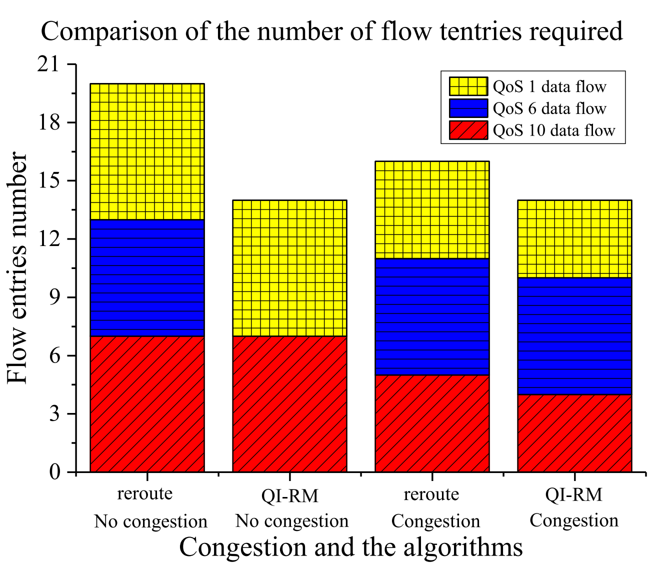


Fig. 15. Number of flow entries required for path change comparison.

5.2.2. Experiments on the QI-RM

In the experiments, since the path selected by QI-RM is modified on the basis of the path selected by ECMP, QI-RM is compared with ECMP and the traditional Dijkstra algorithm.

For QoS 1 data flow, according to its bandwidth requirement, its sending rate was set as 210 kbps and the data quantity was set as 1 MB, and recorded the average throughput of two data flows the flowing to each of the three algorithms in the case of no link congestion and link congestion. In the case of link congestion, in the h1→h6 streaming data flow simulation, the background flow with the sending rate of 890 kbps was added on the link s1–s3, s2–s5 and s1–s5 to generate congestion under three algorithms, respectively. When carrying out a similar experiment on h1→h8 flow, the background flow with the sending rate of 890 kbps was added on the link s1–s5, s2–s5 and s1–s6 to generate congestion under three algorithms, respectively. The average throughput of the three algorithms QoS 1 data flows before and after congestion is shown in Fig. 12. For the h1→h6 QoS 1 data flow, the average throughput of Dijkstra shortest path based on hops, ECMP algorithm and the QI-RM are 139.1 kbps, 137.9 kbps and 209.4 kbps respectively. For the h1→h8 QoS 1 data flow, the average throughput of the three algorithms is 138.1 kbps, 137.8 kbps, and 209.95 kbps, respectively. After the link congestion, the average throughput of the three algorithms is 127.9 kbps, 127 kbps and 206.18 kbps respectively for the h1→h6 QoS 1 data flow. For the h1→h8 QoS 1 data flow, the average throughput of the three algorithms is 126.8 kbps, 126.5 kbps, and 206.76 kbps, respectively. In the case of no congestion, the throughput of the data flow in this paper is above the threshold, and the throughput of the other two algorithms is below the threshold. Therefore, the QI-RM can guarantee the requirement of QoS 1 data flow.

In this paper, the average delay of two data flow streaming to QoS 6 with or without link congestion are also tested. The average delay in the simulation results refers to the average round-trip time. In the case of link congestion, in the h1→h6 data flow simulation, the average delay of the three algorithms was tested. The background flow with the rate of 890 kbps was added on the link s1–s3, s2–s5 and s2–s5 to generate congestion under three algorithms, respectively. When testing the h1→h8 QoS 6 data flow in the case of link congestion, the background flow with the rate of 890 kbps was added on the link s1–s3 to generate congestion under three algorithms, respectively. The average delay of QoS 6 data flow before and after congestion of the three algorithms is shown in Fig. 13. For the data flow of QoS 6 from h1 to h6, the average delay of Dijkstra shortest path based on hops, ECMP algorithm and the QI-RM are 104.54 ms, 104.36 ms and 104.98 ms respectively. For the h1→h8 streaming QoS 1 data flow, the average delay of the three algorithms is 104.84 ms, 104.12 ms and 104.40 ms, respectively. After the link congestion, the average delay of the three algorithms is 262.29 ms, 266.1 ms and 108.54 ms respectively for the h1→h6 QoS 6 data flow. For the h1→h8 QoS 6 data flow, the average delay of the three algorithms is 263.07 ms, 264.68 ms and 108.57 ms, respectively. In no congestion, the delay of the three algorithms is below the threshold. But in the case of congestion, only the delay of the QI-RM is lower than the threshold. Therefore, the QI-RM can guarantee QoS 6 data flow requirements.

The average packet loss rate of QoS 10 with or without link congestion was also tested in this paper. In the case of link congestion, in the h1→h6 data flow simulation, when testing three algorithms, the background flow with the rate of 890 kbps was added on the link s1–s5, s2–s4 and s1–s5 to generate congestion under three algorithms, respectively. In the test of h1→h8 QoS 10 data flow, the background flow with the rate of 890 kbps was added to generate congestion on the link s1–s5, s2–s3 and s1–s5, respectively. The average packet loss rate of the three algorithms QoS 10 data flow before and after congestion is shown in Fig. 14. For the data flow of QoS 10 from h1 to h6, the average packet loss rates of Dijkstra shortest path based on hops, ECMP algorithm and the QI-RM are 2.23%, 1.43% and 0.11% . For the h1→h8

Table 8
Path before and after congestion and comprehensive weight.

Data flow	Algorithms	Dst	Forwarding path (no congestion/congestion)	Forwarding path comprehensive weight (no congestion/congestion)
QoS 1	Dijkstra based on hops	h6	s7→s3→s1→s5→s9/s7→s3→s1→s5→s9	1.348/10001.202
		h8	s7→s3→s1→s5→s10/s7→s3→s1→s5→s10	1.366/10001.220
	ECMP	h6	s7→s3→s2→s5→s9/s7→s3→s2→s5→s9	1.412/10001.242
		h8	s7→s3→s2→s5→s10/s7→s3→s2→s5→s10	1.430/10001.260
	The QI-RM	**h6**	**s7→s4→s1→s5→s9/s7→s4→s2→s5→s9**	**265/0.392**
		h8	**s7→s4→s1→s6→s10/s7→s4→s2→s6→s10**	**0.324/0.326**
QoS 6	Dijkstra based on hops	h6	s7→s3→s1→s5→s9/s7→s3→s1→s5→s9	1.067/ 10000.879
		h8	s7→s3→s1→s5→s10/s7→s3→s1→s5→s10	1.068/10000.880
	ECMP	h6	s7→s4→s2→s5→s9/s7→s4→s2→s5→s9	0.948/10000.736
		h8	s7→s3→s1→s5→s10/s7→s3→s1→s5→s10	1.068/10000.832
	The QI-RM	**h6**	**s7→s4→s2→s5→s9/s7→s3→s1→s5→s9**	**0.946/1.073**
		h8	**s7→s3→s1→s5→s10/s7→s4→s2→s5→s10**	**1.067/0.951**
QoS 10	Dijkstra based on hops	h6	s7→s3→s1→s5→s9/s7→s3→s1→s5→s9	1.934/ 10001.908
		h8	s7→s3→s1→s5→s10/s7→s3→s1→s5→s10	1.935/10001.909
	ECMP	h6	s7→s4→s2→s6→s9/s7→s4→s2→s6→s9	0.625/10001.089
		h8	s7→s3→s2→s6→s10/s7→s3→s2→s6→s10	1.935/10001.563
	The QI-RM	**h6**	**s7→s4→s1→s5→s9/s7→s4→s2→s5→s9**	**0.107/0.625**
		h8	**s7→s4→s1→s5→s10/s7→s4→s2→s5→s10**	**0.108/0.626**

QoS 10 data flow, the average packet loss rate of the three algorithms is 2.22%, 2.65% and 0.12%. After the occurrence of link congestion, the average packet loss rate of the three algorithms is 10.91%, 9.64% and 0.91% respectively for the h1→h6 QoS 10 data flow. For the data flow of QoS 10 from h1 to h8, the average packet loss rate of the three algorithms is 9.8%, 10.83% and 0.92% respectively. With or without congestion, the packet loss rate of the QI-RM is lower than the threshold, while the packet loss rate of the other two algorithms is higher than the threshold. Therefore, the QI-RM can guarantee QoS 10 data flow requirements.

The comparison between the QI-RM and the number of flow entries required by the complete reroute algorithm from source to destination for path change is shown in Fig. 15. Both before and after congestion, the flow table terms required by the QI-RM to guarantee data QoS requirements are lower than the reroute algorithm.

From all the simulation results above, both the Dijkstra shortest path algorithm based on hops and ECMP algorithm based on hops cannot guarantee the QoS requirements of all data flows, while the QI-RM in this paper can guarantee the QoS requirements of all data flows. According to the path results shown in Table 8, Dijkstra shortest path algorithm based on hops and ECMP algorithm do not consider the network link state and QoS requirement of data flow when making path selection, and the path comprehensive weight is large, which makes both algorithms cannot meet QoS requirement of all data flows. The QI-RM considers the network link state and QoS requirements of the data flows obtained by the MACCA2-RF&RF algorithm. According to QoS requirements, the path with lower comprehensive weight can be selected for different data flows to guarantee QoS requirements of different data flows. In accordance with the path results shown in Table 8, the QI-RM can also detect the congestion link after congestion and change the forwarding path of the data flow on the congestion link. Compared with Dijkstra shortest path algorithm based on hops and ECMP algorithm, the QI-RM can guarantee QoS requirements of relevant data flows. And the QI-RM requires less flow table, the main reason is that initial ECMP algorithm calculates link under certain circumstances can meet the QoS requirements, data flow and local routing change algorithm only makes local forwarding path change which makes the QI-RM required fewer flow entries.

6. Conclusion and future work

The generation of massive IoT data of different services makes it more and more difficult for the traditional network to guarantee QoS requirements of data flow. The centralized control of SDN makes it easier to achieve the guarantee of data flow QoS requirements. In order to realize the efficient routing that guarantees QoS in SDN, the QoS requirements of data flow need to be recognized efficiently and accurately first. In this paper, combining multiple machine learning methods, a data flow classification algorithm (MACCA2-RF&RF) is proposed. MACCA2-RF&RF consists of two base classifiers, a misclassification results judgment result module and a decision module. Under the condition of selecting a small number of data packets, MACCA2-RF&RF can quickly and accurately classify data flows to obtain QoS requirements. Considering different QoS requirements, the link parameters and thresholds are newly set. In order to realize the efficient routing that guarantees QoS in SDN, the QoS guaranteed path selection algorithm and local route change algorithm are proposed, which can dynamically adjust the routing path according to the QoS requirements of the data flows and the state of the link, before and after congestion. Based on the above, a QoS guaranteed intelligent routing mechanism in SDN called QI-RM was proposed. According to the simulation results, the QI-RM can provide a path to guarantee QoS for all kinds of data flows and guarantee QoS requirements for all kinds of data flows. In addition, the QI-RM proposed in this paper can be applied not only in IoT, but also in other networks with large number of data flows and various types of QoS requirements, such as 5G and 6G communications.

Although the QI-RM can effectively guarantee the QoS requirements of data flow, it still has a lot of things to be optimized. For example, when changing the forwarding path of data flow, controller still needs to send multiple flow entries, which leads to more touch points in the network, resulting in operation and maintenance difficulties. Combined with the segment routing, controller only needs to communicate with the entrance switch to complete the change of data flow path, reducing the touch point. Therefore, in the future work, we will continue to optimize the QI-RM by combining the segment routing.

CRediT authorship contribution statement

Weifeng Sun: Conceptualization, Methodology, Investigation, Writing - review & editing, Funding acquisition. **Zun Wang:** Conceptualization, Methodology, Investigation, Writing - review & editing. **Guanghao Zhang:** Visualization, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

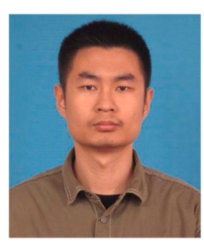
Acknowledgments

This work was supported by the CERNET Innovation Project, China (NGII20190801) and the National Key R&D Program of China (2018YFB1700100).

References

[1] G. Li, L. Boukhatem, J. Wu, Adaptive quality-of-service-based routing for vehicular ad hoc networks with ant colony optimization, IEEE Trans. Veh. Technol. 66 (4) (2017) 3249–3264.
[2] J. Wu, S. Guo, J. Li, D. Zeng, Big data meet green challenges: Big data toward green applications, IEEE Syst. J. 10 (3) (2016) 888–900.
[3] Muhammad Babar, Fazlullah Khan, Waseem Iqbal, Abid Yahya, Fahim Arif, Zhiyuan Tan, Joseph Chuma, A secured data management scheme for smart societies in industrial internet of things environment, IEEE Access 6 (2018) 43088–43099.
[4] R. Amin, M. Reisslein, N. Shah, Hybrid SDN networks: A survey of existing approaches, IEEE Commun. Surv. Tutor. 20 (2018) 3259–3306.
[5] Feilong Tang, Lu Li, Leonard Barolli, Can. Tang, An efficient sampling and classification approach for flow detection in SDN-based big data centers, in: Proceedings - International Conference on Advanced Information Networking and Applications, AINA, 2017, pp. 1106–1115.
[6] Weifeng Sun, Zun Wang, Guanghao Zhang, Boxiang Dong, MACCA: A SDN based collaborative classification algorithm for QoS guaranteed transmission on IoT, in: 15th International Conference on Advanced Data Mining and Applications, ADMA 2019.
[7] Guangyu Li, Lila Boukhatem, Jinsong Wu, Adaptive quality-of-service-based routing for vehicular ad hoc networks with ant colony optimization, IEEE Trans. Veh. Technol. 66 (4) (2017) 3249–3264.
[8] C. Shyamala, M. Geetha Priya, K.A. Sumithra Devi, Cross layer qos guaranteed fault tolerance for data transmission in mobile wireless sensor networks, Wirel. Pers. Commun. (2020) http://dx.doi.org/10.1007/s11277-020-07472-2.
[9] Wile Sehery, Charles Clancy, Flow optimization in data centers with clos networks in support of cloud applications, IEEE Trans. Netw. Serv. Manag. 14 (2017) 847–859.
[10] Guo-Cin Deng, Kuochen Wang, An application-aware QoS routing algorithm for SDN-based IoT networking, in: 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, pp. 186–191.
[11] Joonseok Park, Jeseung Hwang, Keunhyuk Yeom, NSAF: An approach for ensuring application-aware routing based on network QoS of applications in SDN, Mob. Inf. Syst. 2019 (2019) http://dx.doi.org/10.1155/2019/3971598.
[12] L.V. Qi-di, Design and Implementation of Routing Algorithms Module in WAN Traffic Scheduling System Based on SDN (Ph.D. thesis), University, 2017.
[13] Chunzhi Wang, Zhang Gang, Xu Hui, Hongwei Chen, An ACO-based link load-balancing algorithm in SDN, in: International Conference on Cloud Computing & Big Data, 2016, pp. 214–218.
[14] Lei Ming, Li Jing, Fu Yan-fang, Research on flow scheduling algorithm in software defined network, Foreign Electron. Meas. Technol. 10 (2017) 23–26.
[15] Yonggui Wang, H.U. Caiyun, L.I. Xin, School of Software, and Liaoning Technical University, Perturbation particle swarm optimization algorithm based on local far-neighbor differential enhancement, J. Comput. Appl. 38 (2018) 1239–1244.
[16] X.U. Kaibo, L.U. Haiyan, Biyun Cheng, Yang Huang, School of Science, and Jiangnan University, Ant colony optimization algorithm based on improved pheromones double updating and local optimization for solving TSP, J. Comput. Appl. 37 (2017) 1686–1691.
[17] Wang, Likun, Yao, Lin, Xu, Zichuan, Wu, Guowei, Obaidat, Mohammad, CFR: A cooperative link failure recovery scheme in software-defined networks, Int. J. Commun. Syst. 31 (2018) http://dx.doi.org/10.1002/dac.3560.
[18] Haythem Yahyaoui, Saifeddine Aidi, Mohamed Faten Zhani, On using flow classification to optimize traffic routing in SDN networks, in: 2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC), 2020, http://dx.doi.org/10.1109/CCNC46108.2020.9045216.
[19] Gaolei Li, Mianxiong Dong, Kaoru Ota, Jun Wu, Tianpeng Ye, Deep packet inspection based application-aware traffic control for software defined networks, in: Globecom IEEE Global Communications Conference, 2016, http://dx.doi.org/10.1109/GLOCOM.2016.7841721.
[20] Pedro Amaral, Joao Dinis, Paulo Pinto, Luis Bernardo, Henrique S. Mamede, Machine learning in software defined networks: Data collection and traffic classification, in: IEEE International Conference on Network Protocols, 2016, http://dx.doi.org/10.1109/ICNP.2016.7785327.
[21] M.Z. Fatimah Audah, Tan Saw Chin, Y. Zulfadzli, C.K. Lee, K. Rizaluddin, Towards efficient and scalable machine learning-based QoS traffic classification in software-defined network, Lecture Notes in Comput. Sci. 11673 LNCS (2019) 217–229.
[22] Shou-Chieh Chao, Kate Ching-Ju Lin, Ming-Syan Chen, Flow classification for software-defined data centers using stream mining, IEEE Trans. Serv. Comput. 12 (1) (2019) 105–116.
[23] Pan Wang, Feng Ye, Xuejiao Chen, Yi Qian, DataNet: Deep learning based encrypted network traffic classification in SDN home gateway, IEEE Access 6 (2018) 55380–55391.
[24] Jose Suarez-Varela, Pere Barlet-Ros, Flow monitoring in software-defined networks: Finding the accuracy/performance tradeoffs, Comput. Netw. 135 (apr.22) (2018) 289–301.
[25] Pu Wang, Shih Chun Lin, Min Luo, A framework for qos-aware traffic classification using semi-supervised machine learning in SDNs, in: IEEE International Conference on Services Computing, 2016, pp. 760–765.
[26] Hamid Tahaei, Firdaus Afifi, Adeleh Asemi, Faiz Zaki, Nor Badrul Anuar, The rise of traffic classification in IoT networks: A survey, J. Netw. Comput. Appl. 154 (2020) http://dx.doi.org/10.1016/j.jnca.2020.102538.
[27] Li Wei, Marco Canini, Andrew W. Moore, Raffaele Bolla, Efficient application identification and the temporal and spatial stability of classification schema, Comput. Netw. 53 (2009) 790–809.
[28] Yang Kai, Lin Hai-tao, Li Hai-xue, Overview on traffic controlling algorithm based on SDN, Commun. Technol. (2019).

Dr. Weifeng Sun received the PH.D degree and bachelor degree in University of Science and Technology of China (USTC) in 2007 and 2002, respectively. Now he is an associate professor in School of Software, Dalian University of Technology (DLUT). He was a visiting scholar at National Center for Super Computer Applications (NCSA) at University of Illinois at Urbana–Champaign (UIUC) in U.S. (2017–2018), working on the high performance computing on Location Base Service for CyberGIS. Currently, most of his research centers around high quality wireless communications on heterogeneous wireless multi-hop network; intelligent scheduling and cross layer design based on SDN; crowd intelligent algorithms and learning researches on applications in Industrial Internet of Things(IIoT). He serves as Publicity Chair, Session Chair or TPC Member of a number of international conferences. He has authored/co-authored over 80 scientific papers in scientific journals and conference proceedings.



Zun Wang received the B.S. degree in software engineering from Wuhan University of Science and Technology(WUST), Wuhan. China, in 2018. He is currently working toward the Master's degree in software engineering at the School of Software, Dalian University of Technology (DLUT), Dalian, China. He is an excellent graduate student of DLUT and has been awarded several scholarships. His research interests include SDN routing optimization and industrial Internet of Things.

Guanghao Zhang received the B.S. degree in communication engineering from Xidian University (XDU), Xi'an. China, in 2019. He is currently working toward the Master's degree in software engineering at the School of Software, Dalian University of Technology (DLUT), Dalian, China. He is an excellent graduate student of DLUT and has been awarded several scholarships. His research interests include wireless network routing optimization and industrial Internet of Things.