A Single-Hop Selection Strategy of VNFs Based on Traffic Classification in NFV

Bo He(✉), Jingyu Wang, Qi Qi, and Haifeng Sun

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
{hebo, wangjingyu, qiqi, sunhaifeng_1}@ebupt.com

Abstract. Network Function Virtualization (NFV) has become a hot technology since it provides the flexible management of network functions and efficient sharing of network resources. Network resources in NVF require an appropriate management strategy which often manifests as a difficult online decision making task. Resource management in NFV can be thought of as a process of virtualized network functions (VNFs) selection or deployment. This paper proposes a single-hop VNFs selection strategy to realize network resource management. For satisfying quality requirements of different network services, this strategy is based on the results of traffic classification which utilizes Multi-Grained Cascade Forest (gcForest) to distinguish user behaviors on the internet. In the order of VNFs, a network is divided into several layers where each arrived packet needs to queue. The scheduler of each layer selects a layer which hosts the next VNF for the packets in the queue. Experiments prove that the proposed traffic classification method increases the precision by 7.7% and improves the real-time performance. The model of VNFs selection reduces network congestion compared to traditional single-hop scheduling models. Moreover, the number of packets which fail to reach target node in time drops 30% to 50% using the proposed strategy compared to the strategy without the section of traffic classification.

Keywords: NFV · Traffic classification · Resource management · VNFs selection

1 Introduction

The quality of service (QoS) requirements of users rise rapidly such as lower latency, lower packet loss rate and so on. For this reason, networks not only need to enhance bandwidth and capacity, but also require a better scheduling strategy of resources. By separating network functions from traditional hardware, NFV is expected to manage network functions and share network resources more flexibly. Application of NFV has become extensive because more customized network scale and lower capital expenditure are obtained by this technique. In NFV, virtualized network functions (VNFs) which control the creation, configuration, monitor, operation and security of network functions are implemented in software components running on commodity hardware. Services are realized by VNFs in a specific order denominated Service Function Chain

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 267–283, 2019.
https://doi.org/10.1007/978-3-030-12981-1_19

(SFC). In detail, a traffic packet needs to traverse the nodes which host the VNFs in a specific SFC sequentially. As a result, the strategy of resource management is equal to the strategy of selection and deployment of VNFs.

State-of-the-art efforts about network resource management in NFV are limited to optimizing the algorithms in the interior of network for the purpose that all packets arrive the target node as soon as possible [1–3]. To different kinds of traffic packets, the efforts do not realize the QoS requirements are distinguishing. For example, there is a high demand of online video applications for traffic transmission delay, otherwise it will seriously affect the normal use of network services. By contrast, users do not have urgent requirements of delay when they use File Transfer Protocol (FTP) applications, they need lower packet loss rate instead. Under the existing strategies, significant resources may be occupied by traffic packets of FTP instead of video streaming so that users have bad QoS when watching videos. From this issue, it is natural to think about classifying traffic before scheduling network resource. Then resource scheduling problem is considered to be a combinatorial optimization problem. If a packet can be classified before being transmitted, the delay and packet loss rate requirements of this kind of traffic packet are obtained. Thus the network resource management strategy can exploit the requirements to improve user QoS.

This paper proposes a single-hop selection strategy of VNFs based on traffic classification to schedule network resources in NFV. As the premise of VNFs selection, traffic classification needs to identify the transmission priority of different packets accurately. There have been extensive researches of traffic classification, but they were limited in several specific applications [4–6] or unencrypted packets [7, 8]. In order to distinguish user QoS requirements of different network services, we classify the traffic data according to the user behavior. This categorization achieves covering majority of traffic packets in actual network instead of a few applications. Features of classification are calculated by the arrival times, number and lengths of packets to investigate the differences among user behavior. Unlike traditional features, these selected features can be obtained even the packets are encrypted. As for algorithms of classification, this paper tries deep neural network (DNN) to classify traffic data due to some advantages of multi-layer neural network models in the field of data classification. Besides, a new algorithm called Multi-Grained Cascade Forest (gcForest) [9] which is presented as an alternative to DNN is also employed. In the tasks of giving features, gcForest often obtains better results than DNN.

The proposed VNFs selection strategy divides the network into several layers according to the order of VNFs in SFC. In each layer, each arrived packet needs to queue. Then the packet will be transmitted to a next VNF layer which is selected by the scheduler. VNF layers are selected according to the results of traffic classification with joint consideration of the network real-time bandwidth and computing resources. Experiments prove that the traffic classification method gets higher precision than previous work [10] and improves real-time performance of resource management. Meanwhile, simulation results demonstrate that the VNFs selection strategy reduces the number of packets which fail to meet the QoS requirement under different degrees of network congestion.

2 Related Work

2.1 Traffic Classification

Traffic classification is a hot issue in academic all the time. Instead of many methods based on the port numbers or the payload data of traffic packets in history, more and more researches employ algorithms of machine learning in the last decade. Williams et al. [4] extracted 22 practical flow features for use within IP traffic classification and employed five algorithms of machine learning to classify traffic. Dong et al. [5] selected four flow features for traffic classification and obtained the accuracy up to 95%. But these features only worked well in classifying six kinds of video traffic which the authors selected. Since previous studies only worked offline, Bernaille et al. [7] proposed a method to classify traffic online by observing the first five packets of each TCP connection. But limitation is that the method classifies only several specific TCP applications. Shi et al. [8] realized accurate classification of several kinds of protocol traffic data by means of complex methods to extract features from traffic flows and remove the irrelevant and redundant features later on. Anderson et al. [11] aimed at overcoming two limitations of detection of malicious network traffic: inaccurate ground truth and highly non-stationary data distribution. An enhanced feature set is presented based on the information of Transport Layer Statistics (TLS) sessions.

Most of existing methods to classify traffic employed the statistics algorithms of machine learning like KNN, SVM, decision tree and so on. Some of the latest studies [6, 12] started to utilize some algorithms of deep learning to classify traffic. However, these studies trained features which were still extracted beforehand so they did not take advantage of the ability of neural network models to extract optimal features. Consequently, these studies do not archive overwhelming advantages over the previous ones.

2.2 Network Resource Management in NFV

Purposes of most existing network resource management strategies is to reduce the number of equipment or improve QoS. Tseng et al. [13] carried out several sets of experiments to prove that selecting suitable discontinuous reception parameters can effectively reduce power consumption of nodes. Joe et al. [14] proposed an algorithm of network selection based on Analytic Hierarchy Process (AHP) for predicting power consumption of terminal equipment in the network. Senouci et al. [15] selected suitable network interfaces in a dynamically changing network by utilizing Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS). Park et al. [16] introduced game theory into the field of resource management and proved that different applications can share effective bandwidth by cooperative game. According to whether traffic packets are sensitive to latency or not, Afzal et al. [17] divided them into two classes for scheduling.

In NFV, some studies proposed optimized scheduling algorithms of typical unicast issue by focusing on computing resources of nodes [18] and bandwidth resources of edges [19]. Sun et al. [1] presented a framework which enables network function work in parallel and it reduced latency greatly for real world service chains. More and more studies started to focus on the impact of VNFs on scheduling resources of network. Taleb et al. [2] calculated loss when a VNF breaks down by estimating the number of active/idle user equipment and proposed a network architecture enabled by service resilience-aware mechanisms. For minimizing latency of end-to-end service, Chantre et al. [3] employed the particle swarm optimization technique to solve the redundancy allocation problems. Mestres et al. [20] initiated experiments to prove different VNFs have different curves of resource consumption with the increase of network data even in the same network. Kar et al. [21] tried to solve the problem that optimizing energy-cost with capacity and delay as constraints so that a dynamic placement of SFC heuristic solution was proposed. Gu et al. [22] proposed an algorithm of placement of VNFs for reducing the communication cost with joint consideration of network flow balancing and predetermined network service semantics.

As mentioned above, most of methods to classify traffic are limited to several applications or protocols. Many features of classification are difficult to be extracted from message format information of traffic packets since more and more packets are encrypted. In NFV, many studies about scheduling resources did not realize that QoS requirements of traffic packets are different. Thus, they never thought of utilizing traffic classification technology to identify the transmission priority of different packets.

3 Architecture Overview

In NFV, VNFs need to be deployed in a specific SFC sequentially (e.g. network address translation function requires postprocessing after firewall function). Therefore, selecting and deploying VNFs already become the methods to schedule resources in NFV. For satisfying QoS requirements of different traffic packets better in existing networks, this paper proposes a selection strategy of VNFs based on traffic classification in NFV instead of a strategy of the VNFs placement. However, actual networks are complex because some nodes host one or more VNFs while the others do not. Even the same VNF may be hosted in different nodes. Thus this strategy adopts a fine-gained single-hop mode to cope with complex and variable actual NFV networks. We combine a section of traffic classification at the source node of the network with a section of VNFs selection in the interior of the network. The premise of employing this VNFs selection strategy is that all the VNFs we need are already deployed according to the order in SFC. That is to say, there is at least one path which satisfies the SFC from the source node to the destination node. The architecture of this system is shown in Fig. 1 and described as follows.

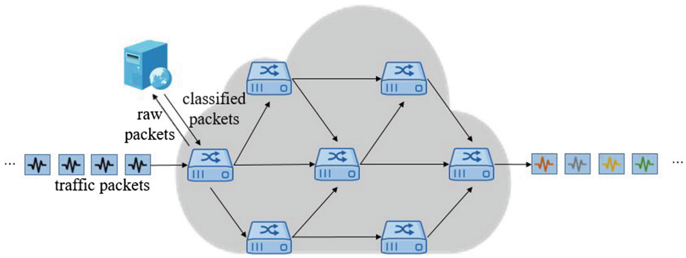


Fig. 1. Illustration of the single-hop selection strategy of VNFs based on traffic classification

Firstly, packets arriving at the source node are sampled and the features of them are extracted and transmitted to an associated server which has a trained model of traffic classification inside. This model distinguishes traffic packets into 8 classes according to user behavior characteristics and QoS requirements, i.e. Browsing, Chat, Audio-streaming, Video-streaming, Email, VoIP, P2P and FTP. These 8 classes are not limited to several specific applications or protocols so that they cover majority of traffic packets in actual networks. This classification coverage is the basis of resource management.

Secondly, the model classifies the input packets and verifies transmission requirements like delay and packet loss rate. According to the classification results, the Differentiated Services Code Point (DSCP) field of every packet is modified as a label.

After being labeled, flow information of the packets is recorded, such as their five-tuple (source IP address, destination IP address, source port number, destination port number, protocol). Then if a new packet which has the same five-tuple is transmitted to the source node, it is labeled as previous ones of this flow and scheduled in this network directly without additional need to be classified.

We emphasize that the transmission time T_i of packet i is obtained by adding the delay time D_e of each edge e and the waiting time W_n on each node n. Let the set E_i represents the set of edges and the set N_i represents the set of nodes which are in the path of packet i passing. During the transmission process, packet i is always checked whether it can continue to be transmitted according to the time t. Thus we have the objective function and the constraint function:

$$T_i = \sum\nolimits_{e \in E_i} D_e + \sum\nolimits_{n \in N_i} W_n \tag{1}$$

$$t \geq s_i + d_i \tag{2}$$

where s_i is the start time of the transmission and d_i is the longest transmission time obtained by the QoS requirement. The scheduler of each layer verifies the priority of transmission of each arrived packet by the label in DSCP field. Then this scheduler selects a layer which hosts the next VNF for the packet. The selection process is repeated until the packet reaches the target node.

4 Methodology

4.1 Traffic Classification

On account of that the features selected by many methods only work well in a specific network environment, we extract some features which are calculated by the information of number, lengths and arrival times of packets instead of traditional message format information like protocol and port number. These features can be extracted regardless of whether traffic packets are encrypted. More importantly, traffic classification can identify traffic packets from different applications as the same class by the features. For example, skype and facebook are different applications but both have the VoIP function to generate two-way traffic packets which have short inter arrival times and similar lengths. As a contrast, traffic packets of FTP also have short inter arrival times, but lengths of packets sent forwards are much bigger than packets sent backwards. According to experiments, this paper sets 5 s as the length of flows. The flows consist of two-way traffic packets that have the same five-tuple. Lashkari et al. [10] classified traffic packets by 23 flow features which only based on time. Table 1 exhibits 13 features which we select from these 23 features. Furthermore, considering the bandwidth requirement in the field of resource management and the differences in user behavior, we extract 18 flow features based on number and lengths of packets and show them in Table 2.

Table 1. Features based on time

Basic data	Features
Packets sent forwards	Inter arrival time (mean, min, max, std)
Packets sent backwards	Inter arrival time (mean, min, max, std)
Packets sent in either direction	Inter arrival time (mean, min, max, std)
Flows	Duration

Table 2. Features based on number and lengths of packets

Basic data	Features
Packets sent forwards	Length (mean, min, max, std), bytes and number per second
Packets sent backwards	Length (mean, min, max, std), bytes and number per second
Packets sent in either direction	Length (mean, min, max, std)
Packets sent forwards and packets sent backwards	Ratio of the number of bytes per second, ratio of the number per second

In order to find an optimal classification algorithm, not only common machine learning algorithms such as decision tree, random forest, KNN and SVM, we also employ DNN and gcForest using the same dataset. Compared to common machine learning algorithms, DNN and gcForest performs better in experiments. Therefore, the principles of the two algorithms are elaborated as follows.

Deep Neural Network (DNN). The DNN is especially the neural network with fully connected layers. DNN is an algorithm of supervised learning, which utilizes the fitting function to realize the classification of input data. Each neural node of DNN learns a linear function according to the weight w and the bias b:

$$z = \sum_{i=1}^{n} w_i x_i + b \tag{3}$$

where n is the number of inputs that the neural node receives and x is the value of the corresponding input. Then the result z is input into an activation function for learning nonlinear data better. The activation function we select is Rectified Linear Unit (ReLU):

$$\mathrm{ReLU}(z) = \begin{cases} z, & z > 0 \\ 0, & z \leq 0 \end{cases} \tag{4}$$

Neural network layers of DNN are divided into three kinds: input layer, hidden layer and output layer. Input layer is used to receive data input to the neural network. Hidden layers can have multiple layers to enhance expressive power of this model. Generally, neural networks with more hidden layers and neural nodes are able to fit more complicated functions. Output layer has multiple output nodes to output predicted results that match the classes of input data. The experiments use five layers to build the model.

The training process of DNN constantly adjusts weight and bias of each neural node to fit the classification function that meets input data better. The evaluation indicator of performance is loss function which we select is cross-entropy cost function:

$$C = -\frac{1}{n}\sum_{i=1}^{n} y \ln a + (1-y)\ln(1-a) \tag{5}$$

where y is the expected output result and a is the label. Labels are input into the model with training data as actual output results. Besides, n is the number of outputs and C is the result of loss. Therefore, loss function indicates the gap between predicted results and the actual results. As for optimization function, we select Adaptive Moment Estimation (Adam) to reduce the value of this loss function for obtaining optimal hyper-parameters. Learning rate determines the speed of modifying the parameter to the better value and has a great impact on the performance of neural network. The learning rate we selected is 0.001.

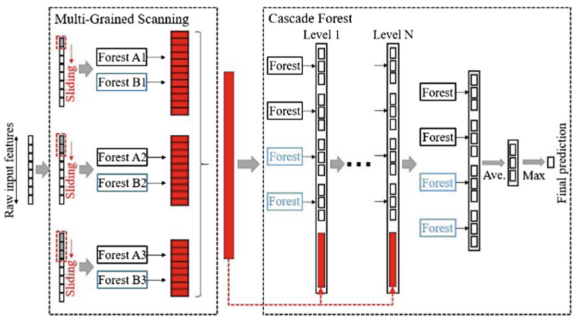


Fig. 2. Architecture of gcForest (Color figure online)

Multi-grained Cascade Forest (gcForest). Multi-Grained Cascade Forest is the improved model of Random Forest [23], which is proposed as an alternative to deep neural networks because its performance is highly competitive in a broad range of tasks. The structure of gcForest is shown in Fig. 2.

The gcForest model consists of two parts: the multi-grained scanning and the cascade forest structure. In the part of multi-grained scanning, sliding windows are used to scan all raw features for forming feature vectors. These feature vectors are used to train completely-random tree forests and random forests to obtain class vectors which are concatenated as transformed features. Random forests contain trees which are generated by choosing the one with the best *gini* value from some randomly selected features for split. And completely-random tree forests consist of regular decision trees. In the part of cascade forest structure, each level composed of different decision tree forests to encourage the diversity which is important to ensemble construction. For example, each level in this structure consists of two random forests (black) and two completely-random tree forests (blue) in Fig. 2. Each level receives information from the preceding level and processes the information by their own forests. Then this level outputs its results to the next level.

Compared to deep neural networks which rely on hyper-parameter tuning, gcForest is much easier to train. In many cases, it works well even using almost same setting of hyper-parameters. The training process of this algorithm is efficient, and users can control training cost according to computational resources available. Moreover, the greatest advantage of gcForest is that the algorithm can obtain a good result in the case of small-scale training data.

4.2 VNFs Selection Strategy

Figure 3 presents an example network in NFV to explain the mechanism of the VNFs selection strategy. A circle with a text represents a node which hosts a VNF and a circle with no text represents a node which only transmits traffic packets. The red and blue lines respectively represent a path which satisfies the order of VNFs in SFC from the source node to the target node. A circle of black dotted lines represents a node which is

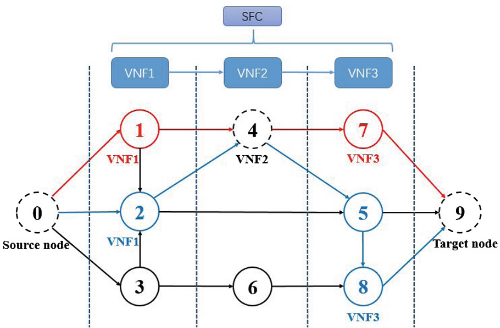


Fig. 3. An example network in NFV. Traffic messages pass through some nodes following the order in SFC from the source node 0 to the target node 9. (Color figure online)

passed through by the both paths. It can be seen that node 0 is the source node and node 9 is the target node.

SFC Layered (SFCL) Model. Considering the characteristic of NFV and the cooperation with traffic classification, a network is divided into layers according to the order of VNFs in SFC. As shown in Fig. 3, traffic packets which are transmitted from source node 0 to target node 9 need to follow the order of VNF1, VNF2 and VNF3 because of the requirement of SFC. Therefore, the network is divided into 5 layers: the source layer, the VNF1 layer, the VNF2 layer, the VNF3 layer and the target layer. Accordingly, the Path1 is divided into node 0, node 1, node 4, node 7, node 9 and the Path2 is divided into node 0, node 2, node 4, node 5, node 8, node 9. What worth mentioning is that node 5 and note 8 are in the same layer in Path2.

Each layer in the network is regarded as an instance object which is represented by a seven-tuple: $\{U, N, X, B, C, K, M\}$. In the seven-tuple, U represents the set of all upper layers of this layer, N represents the set of all next layers of this layer, and X represents the order number of this layer in the path. Besides, B represents the free bandwidth set of edges which connect this layer with its next layers. C represents the congestion degree set of next layers, that is, the numbers of messages waiting to be transmitted in these next layers. Last, K represents a switch that controls this layer whether receives a message from upper layers, M represents the set of messages waiting to be transmitted in the layer. Particularly, an object which has the same index in N, B, C put into correspondence with the same next layer object.

Each message in the network is regarded as an instance object which is represents by a five-tuple: $\{D, S, E, P, R\}$. In the five-tuple, D represents the traffic packet carried by the message, S represents the time when the message enters the network, and E represents the longest transmission time obtained by the result of traffic classification. In addition, P represents the transmission priority of the message, and R represents the probability that the message is transmitted to the next layer which has the best congestion condition.

Scheduling in a Single Layer. Figure 4 shows the scheduling process in a single layer. When a layer receives a message from its upper layer, it first checks whether the time at the moment has exceeded the sum of S and E of the message or not. If the time is exceeded, it means that the message does not reach the target layer within the time delay allowed and it is discarded directly, otherwise the message is put in M of the layer. Then messages of M are sorted by the values of their transmission priorities (P) so that the message which has the largest P is transmitted to one of the next layers. The value of P of message m in layer l is represented by (6).

$$P_m = P_0 + \alpha(E_m + S_m - t) - \beta(L - X_l) \quad (6)$$

where P_0 is the initial transmission priority as the result of traffic classification and t is the time at the moment. L is the number of layers in this network. The values of α and β are determined by conditions of the network so that the scheduler can consider the possibility of this message successfully reaching the target layer in time. If N of the layer has several next layers, each next layer n calculates its own selection priority called SP using (7) in the light of its free bandwidth b_n and congestion degree c_n.

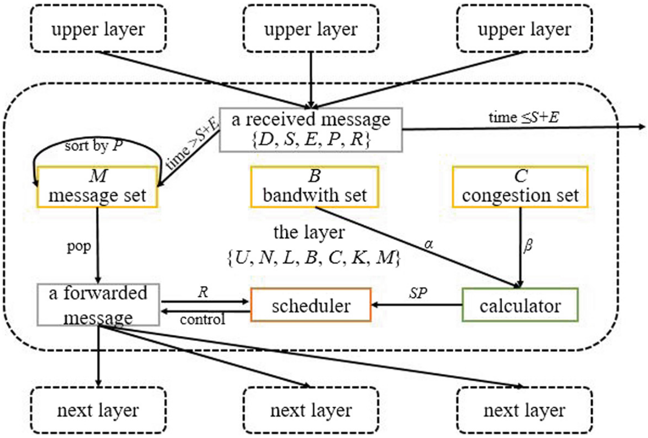


Fig. 4. Scheduling process in a single layer

$$SP = \gamma b_n + \delta c_n \tag{7}$$

where γ and δ are weights of free bandwidth and congestion degree, and their values are determined through experiments. Then the next layers are ranked from high to low according to the values of their *SP*s. At this time, *R* of the message is the probability that it being transmitted to the next layer which has the largest *SP*. Random selection is made according to the 0-1 distribution that matches the value of *R*. The distribution function of the calculated result *x* is:

$$P\{x = k\} = p(1 - p)k, \ (k = 0, 1) \tag{8}$$

where *p* equals to the value of *R*. Thus the probability of different selection results is:

$$P\{x = 0\} = R \tag{9}$$

$$P\{x = 1\} = 1 - R \tag{10}$$

If *x* equals 0, the message will be transmitted to this next layer, otherwise the scheduler will continue to make the random selection to decide whether to transmit this message to the next one in the rank of next layers. By that analogy, if only the last next layer is left, the message is transmitted to this remaining next layer directly.

Considering the interference problem in the process of transmitting messages, it is necessary to ensure that each layer receives only one message from its upper layers at any time. Upper layers of the same layer are mutually called interferential layers. After an upper layer of this layer transmits a message to the layer, the switch *K* is turned off so that interferential layers of the upper layer are banned from transmitting messages to this layer. When the messages are sorted out, *K* is turned on and the layer continues to receive messages. For example, the layer contains node 1 and the layer contains node 2 are interferential layers because they are both upper layers of the layer contains node 4 in Fig. 3.

5 Experiment and Simulation

5.1 Traffic Classification

We carry out experiments about traffic classification using packets of an open dataset [10]. Firstly, we analyze the impact of lengths of traffic flows on traffic classification. To evaluate the classification performance of different algorithms, we use two metrics: precision (PR) and recall (RC). In [10], authors carried out experiments and obtained the results that 15 s is the optimal length and the highest classification precision is 84.1%. Table 3 exhibits the performance of each algorithm under different lengths of the flows. In addition to SVM, the remaining algorithms get high precisions. The results show that 15 s is also the optimal length, but the performance is close under the same algorithm when employing the new features. Thus traffic classification is no longer restricted by the lengths of flows. Considering the real-time performance of network resource management, we choose 5 s as the length of flow without reducing much classification precision. And the gcForest is the optimal algorithm for traffic classification by these new flow features according to the results in Table 3.

Table 3. Results of length selection in experiments

	gcForest		Random forest		Decision tree	
	PR	RC	PR	RC	PR	RC
5 s	0.9179	0.9164	0.9043	0.9083	0.8839	0.8880
10 s	0.9161	0.9134	0.9069	0.9110	0.8870	0.8914
15 s	0.9190	0.9167	0.9072	0.9119	0.8860	0.8896
20 s	0.9213	0.9205	0.9061	0.9097	0.8858	0.8871
	KNN		SVM		DNN	
	PR	RC	PR	RC	PR	RC
5 s	0.8906	0.8959	0.7594	0.7186	0.8868	0.8830
10 s	0.8866	0.8901	0.7980	0.7272	0.8872	0.8834
15 s	0.8856	0.8919	0.7938	0.7153	0.8836	0.8822
20 s	0.8889	0.8930	0.7555	0.7171	0.8966	0.8933

Figure 5 indicates the performance of the six algorithms under different feature sets. We select three feature sets which are shown in Table 4. It can be seen that the precision of classification obtained by feature set1 is 10% lower than the precision of classification obtained by feature set2. These results prove that when the flows are short, it is correct to add the flow features based on number and lengths of packets to the feature set of classification. At the same time, they also show that the precisions of different feature sets from gcForest is highest. Therefore, gcForest is the best classification algorithm even the feature set changes.

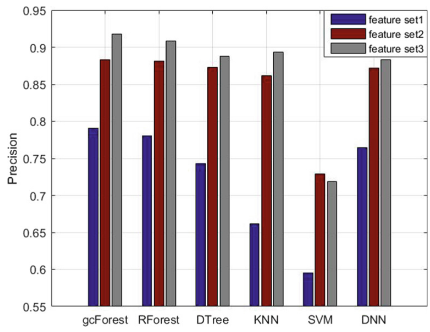
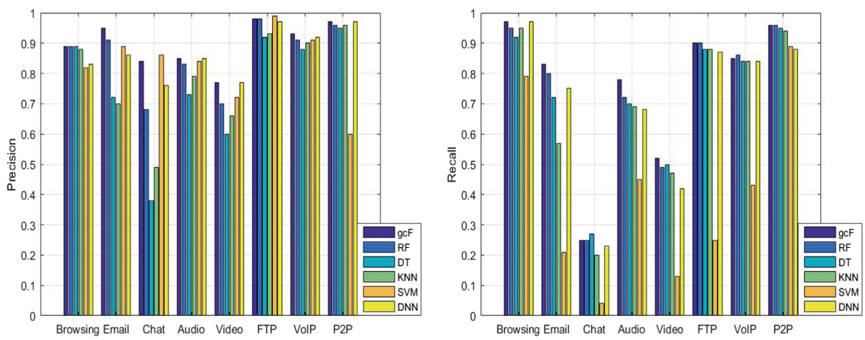


Fig. 5. Precision of different algorithms using different features

Table 4. Different features set

Feature set	Description
Feature set1	Features only based on time
Feature set2	Features based on lengths and number of packets
Feature set3	All features

Figure 6(a) and (b) exhibit the test precisions and recalls of all kinds of traffic. The results show that gcForest and DNN obtain the best performance of recognizing each class of traffic. SVM recognizes several classes of traffic accurately but its overall performance is worse than gcForest. The performance of the traffic classification is satisfactory except some packets originally belong to Chat are wrongly identified as Browsing traffic because some online chat applications run in browsers. Accordingly, the division of traffic classes remains to be optimized.



(a) Precision (b) Recall

Fig. 6. Precision and recall of different kinds of traffic

As for the extra workload of the model with the section of traffic classification, we carry out experiments to evaluate the its impact on real time data transmission. We randomly select 1000 flows from the traffic dataset and measure the total testing time of feature calculation and prediction in the trained classification models. Each flow lasts for 5 s, consisting of the two-way traffic packets which have the same five-tuple. The testing times and precisions of the six trained classification models are shown in Fig. 7. It can be seen that the models classify traffic data quickly and four of them only spend less than 100 ms on 1000 flows. Thus in actual networks, the classification time of a flow is negligible. The proposed scheduling model is feasible due to the section of traffic classification has little impact on real time data transmission.

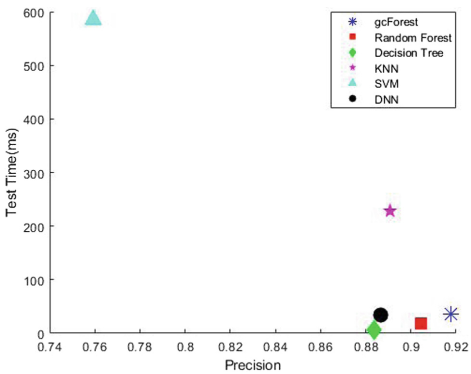


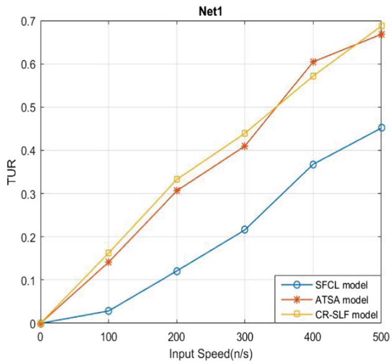
Fig. 7. The testing times and precisions of these classification models using 1000 flows

5.2 VNFs Selection

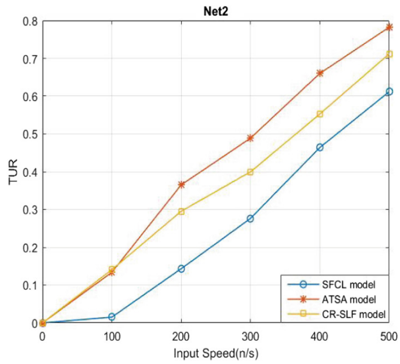
Considering the diversity and complexity of real networks, we select three networks with different topology: Net1, Net2 and Net3. The networks respectively contain 15, 30, 45 nodes and these three all have some nodes to host VNFs. Two nodes of each network are selected as the source node and the target node. In all three networks, there are several paths which have different bandwidth and congestion from the source node to the target node. And the nodes of these paths are guaranteed to satisfy the order of VNF in SFC. We put data into the source nodes of these networks to simulate the selection process of VNFs.

ATSA [24] and CR-SLF [25] are single-hop models of resource management and they both schedule resources according to the longest transmission time of messages, so they are comparable to SFCL. Differently, SFCL is proposed for working based on the results of traffic classification and it takes the bandwidth and congestion of network into consideration. We made experiments to compare the scheduling performance of the three models in each network. In different network, first we evaluate the length of paths and set longest transmission times of different traffic classes to ensure that all packets have chance to reach the target node. For realizing better user QoS, longest transmission times (LTT) of class with higher transmission priority is shorter. In general, the 8 classes of traffic packets are sorted according to the priority of transmission as a specified order, i.e. VoIP, Video-streaming, Audio-streaming, Chat, Browsing, Email, P2P, FTP. Then the parameters of the model are modified by the topology of each network, such as the coefficients in (6) and (7). Finally, we change speed of data input to cause different degrees of congestion in networks and count the number of packets successfully reached the target node in unit time.

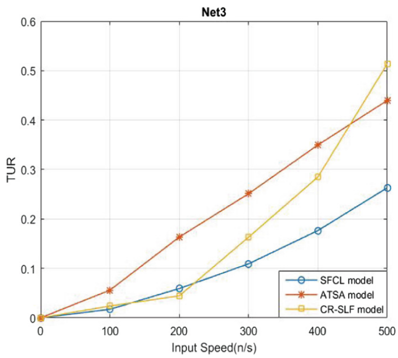
Figure 8(a), (b) and (c) respectively indicate the comparison of scheduling performance obtained by different models in Net1, Net2 and Net3. Experimental message set consists of the 8 classes of messages, each class with 1000 messages. SFCL model works without the section of traffic classification in these experiments. The unlabeled messages are input to the networks in random order. The evaluation indicator of scheduling performance is the Task Unfinished Ratio (TUR). TUR is defined as the ratio of the number of packets that fail to reach the target node before their LTT. The x-axis denotes the speed of data input, namely, the number of messages input per second. And the y-axis denotes that TURs obtained by three models in each network. According to the results, SFCL without section of traffic classification can greatly reduce the TUR compared to the traditional single-hop scheduling models in different network congestion conditions.



(a) TUR in Net1

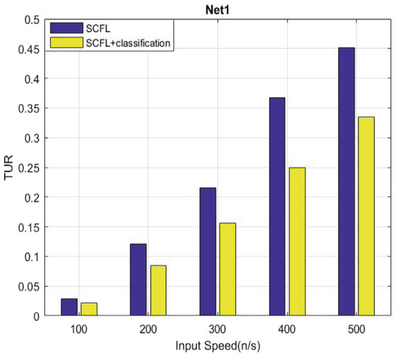


(b) TUR in Net2

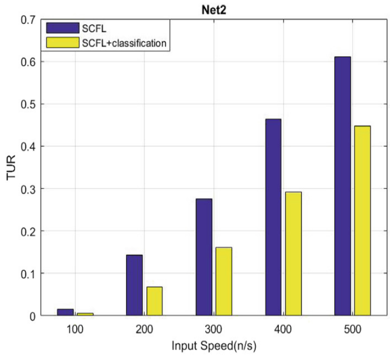


(c) TUR in Net3

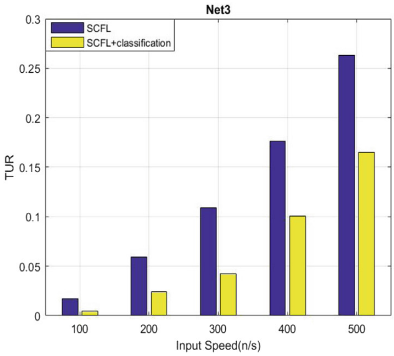
Fig. 8. TURs in different networks under different models



(a) TUR in Net1



(b) TUR in Net2



(c) TUR in Net3

Fig. 9. TURs obtained by adding traffic classification process

The comparisons of the scheduling performance of SFCL model whether to add the section of traffic classification in Net1, Net2 and Net3 are respectively shown in Fig. 9 (a), (b) and (c). Experimental message set is same as before but the messages are labeled after the messages pass through a trained model for classification. Then some parameters of messages are changed for differentiated scheduling, such as transmission priorities. According to the results, TRUs of SFCL decrease by 30% to 50% in different networks after classifying the input traffic packets. Therefore, adding the section of traffic classification is helpful to reduce the TRU and improves user QoS though sometimes the process of classification takes a little time.

6 Conclusion

In this paper, a VNFs selection strategy is proposed to address the problem of scheduling network resources in NFV. The proposed strategy based on a method of traffic classification. This method leverages some flow features based on times, lengths and number of packets to classify different user behaviors on the internet. Results show that the method of classification is no longer restricted by the lengths of flows so that the real-time performance of the resource scheduling strategy is improved. Furthermore, resource management strategy can schedule traffic packets from different user behavior to improve QoS since the method obtains high precision of classification. The gcForest algorithm performs better than other classification algorithms in this task.

The VNFs selection model SFCL proposed by this paper divides the network into layers according to the order of VNFs in SFC. The scheduler of each layer selects a next layer for messages with joint consideration of transmission priority and network congestion. Experiments prove that SFCL can reduce the network congestion effectively compared with the traditional single-hop scheduling models. After adding the section of traffic classification, the number of packets that fail to reach the target node before longest transmission time is reduced by 30% to 50%.

Acknowledgment. This work was jointly supported by: (1) National Natural Science Foundation of China (No. 61771068, 61671079, 61471063, 61372120, 61421061); (2) Beijing Municipal Natural Science Foundation (No. 4182041, 4152039); (3) the National Basic Research Program of China (No. 2013CB329102).

References

1. Sun, C., Bi, J., Zheng, Z., et al.: NFP: enabling network function parallelism in NFV. In: Proceedings of the Conference of the ACM Special Interest Group on Data Communication, pp. 43–56. ACM (2017)
2. Taleb, T., Ksentini, A., Sericola, B.: On service resilience in cloud-native 5G mobile systems. IEEE J. Sel. Areas Commun. **34**(3), 483–496 (2016)
3. Chantre, H.D., da Fonseca, N.L.S.: Redundant placement of virtualized network functions for LTE evolved Multimedia Broadcast Multicast Services. In: 2017 IEEE International Conference on Communications, ICC 2017, pp. 1–7. IEEE (2017)
4. Williams, N., Zander, S., Armitage, G.: A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Comput. Commun. Rev. **36**, 5–16 (2006)
5. Dong, Y.N., Zhao, J.J., Jin, J.: Novel feature selection and classification of internet video traffic based on a hierarchical scheme. Comput. Netw. **119**, 102–111 (2017)
6. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A., et al.: Network traffic classifier with convolutional and recurrent neural networks for Internet of Things. IEEE Access **5**(99), 18042–18050 (2017)
7. Bernaille, L., Teixeira, R., Akodkenou, I., et al.: Traffic classification on the fly. ACM SIGCOMM Comput. Commun. Rev. **36**(2), 23–26 (2006)
8. Shi, H., Li, H., Zhang, D., et al.: Efficient and robust feature extraction and selection for traffic classification. Comput. Netw. Int. J. Comput. Telecommun. Netw. **119**(C), 1–16 (2017)
9. Zhou, Z.H., Feng, J.: Deep forest: towards an alternative to deep neural networks. In: International Joint Conference on Artificial Intelligence, pp. 3553–3559 (2017)
10. Lashkari, A.H., Gil, G.D., Mamun, M.S.I., et al.: Characterization of tor traffic using time based features. In: International Conference on Information Systems Security and Privacy, pp. 253–262 (2017)

11. Anderson, B., Mcgrew, D.: Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. In: The ACM SIGKDD International Conference, pp. 1723–1732. ACM (2017)
12. Shi, H., Li, H., Zhang, D., et al.: An efficient feature generation approach based on deep learning and feature selection techniques for traffic classification. Comput. Netw. **132**, 81–98 (2018)
13. Tseng, C.C., Wang, H.C., Kuo, F.C., et al.: Delay and power consumption in LTE/LTE-A DRX mechanism with mixed short and long cycles. IEEE Trans. Veh. Technol. **65**(3), 1721–1734 (2016)
14. Joe, I., Kim, W.T., Hong, S.: A network selection algorithm considering power consumption in hybrid wireless networks. IEICE Trans. Commun. **91**(1), 1240–1243 (2007)
15. Senouci, M.A., Mushtaq, M.S., Hoceini, S., et al.: TOPSIS-based dynamic approach for mobile network interface selection. Comput. Netw. **107**(2), 304–314 (2016)
16. Park, H., van der Schaar, M.: Bargaining strategies for networked multimedia resource management. IEEE Trans. Signal Process. **55**(7), 3496–3511 (2007)
17. Afzal, B., Alvi, S.A., Shah, G.A., et al.: Energy efficient context aware traffic scheduling for IoT applications. Ad Hoc Netw. **62**, 101–115 (2017)
18. Lukovszki, T., Schmid, S.: Online admission control and embedding of service chains. In: Scheideler, C. (ed.) Structural Information and Communication Complexity. LNCS, vol. 9439, pp. 104–118. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25258-2_8
19. Jia, M., Liang, W., Huang, M., et al.: Throughput maximization of NFV-enabled unicasting in software-defined networks. In: 2017 IEEE Global Communications Conference, GLOBECOM 2017, pp. 1–6. IEEE (2017)
20. Mestres, A., Rodrigueznatal, A., Carner, J., et al.: Knowledge-defined networking. ACM SIGCOMM Comput. Commun. Rev. **47**(3), 2–10 (2016)
21. Kar, B., Wu, H.K., Lin, Y.D.: Energy cost optimization in dynamic placement of virtualized network function chains. IEEE Trans. Netw. Serv. Manag. **PP**(99), 1 (2017)
22. Gu, L., Tao, S., Zeng, D., et al.: Communication cost efficient virtualized network function placement for big data processing. In: Computer Communications Workshops, pp. 604–609. IEEE (2016)
23. Breiman, L.: Random forest. Mach. Learn. **45**, 5–32 (2001)
24. Lee, J., et al.: A real-time message scheduling scheme based on optimal earliest deadline first policy for dual channel wireless networks. In: Sha, E., Han, S.-K., Xu, C.-Z., Kim, M.-H., Yang, L.T., Xiao, B. (eds.) EUC 2006. LNCS, vol. 4096, pp. 264–273. Springer, Heidelberg (2006). https://doi.org/10.1007/11802167_28
25. Li, H., Shenoy, P., Ramamritham, K.: Scheduling messages with deadlines in multi-hop real-time sensor networks. In: IEEE Real Time on Embedded Technology and Applications Symposium. pp. 415–425. IEEE Computer Society (2005)