Application Identification for Virtual Reality Video with Feature Analysis and Machine Learning Technique

Xiaoyu Liu(✉), Xinyu Chen, Yumei Wang, and Yu Liu

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
{liuxiaoy, chenxinyu, ymwang, liuy}@bupt.edu.cn

Abstract. Immersive media services such as Virtual Reality (VR) video have attracted more and more attention in recent years. They are applications that typically require large bandwidth, low latency, and low packet loss ratio. With limited network resources in wireless network, video application identification is crucial for optimized network resource allocation, Quality of Service (QoS) assurance, and security management. In this paper, we propose a set of statistical features that can be used to distinguish VR video from ordinary video. Six supervised machine learning (ML) algorithms are explored to verify the identification performance for VR video application using these features. Experimental results indicate that the proposed features combined with C4.5 Decision Tree algorithm can achieve an accuracy of 98.6% for VR video application identification. In addition, considering the requirement of real-time traffic identification, we further make two improvements to the statistical features and training set. One is the feature selection algorithm to improve the computational performance, and the other is the study of the overall accuracy in respect to training set size to obtain the minimum training set size.

Keywords: Application identification · Statistical feature · Machine learning VR video application

1 Introduction

Nowadays, online video has become one of the most popular network services, and video traffic is increasing on a large scale. For ordinary video, delay or stalling will reduce the Quality of Experience (QoE) of users. Delay or stalling can even cause users' physiological discomfort for Virtual Reality (VR) video. Therefore, it is necessary to establish an effective identification system for VR video application to manage network resources. To the best of our knowledge, there are few studies related to VR video traffic identification. Therefore, we survey several popular methods of network traffic identification and analyze their identification performance for VR video.

Typically, there are four different kinds of methods for network traffic identification, i.e., port-based, host-behavior-based, payload-based, and machine learning (ML) -based. The *port-based* method checks the port number of each packet and compares it with the Internet Assigned Numbers Authority (IANA) list [1]. The IANA

X. Liu et al. (Eds.): ChinaCom 2018, LNICST 262, pp. 336–346, 2019.
https://doi.org/10.1007/978-3-030-06161-6_33

list characterizes the one-to-one relationship between the port number and application. The *host-behavior-based* approach analyzes the host-behavior pattern of the transport layer and then associates the host-behavior with one or more application types [2]. Kim et al. [3] proved that the method based on port and host-behavior were not suitable for video identification.

Initially, we try to identify VR video with the *payload-based* method. Payload-based method checks if the payload of the packet contains a pre-registered special application sequence which is associated with one or more application types. We attempt to search a special application sequence that can distinguish VR video from ordinary video from the following three aspects, i.e., the specific host domain name included in the request packet, the specific video extension name, and the specific content type in the reply packet. However, there are no new discoveries. Therefore, we determine to distinguish VR video from ordinary video with a *ML-based* method. ML can classify each traffic flow by using its statistical features. We use analysis and traffic capture method to obtain statistical features, e.g., average packet size, throughput, packet arrival interval, which can be used to distinguish VR video from ordinary video. Experiments show that using these features, VR video can be well identified. To improve identification speed, we make two improvements to the statistical features and training set, i.e., reducing feature numbers and minimizing training set size.

The remainder of this paper is structured as follows. Section 2 presents background. Section 3 proposes VR video application identification system. And our experimental results are presented in Sect. 4. Finally, Sect. 5 concludes the paper.

2 Background

2.1 Related Work

As far as we know, currently there are few papers related to VR video traffic identification. Most of the work is the categorization of various types of network traffic. For example, karagiannis et al. [2] classified traffic into Web, News, Streaming, Gaming, etc. Moreover, authors in [4, 5] divided the flow data into video and non-video. They compared the performance obtained by Random Forest and AdaBoost, respectively. The results showed that, ignoring the classification speed of the model, the two algorithms could achieve similar classification accuracy (about 93%). Random Forest could guarantee a smaller model while ensuring classification accuracy, leading to faster classification speed. This is also what we obtain during our experiments.

Moore et al. [6] proposed the definition and the calculation of 249 flow features. Later researchers who use statistical features to classify traffic flow will generally adopt a subset of these flow features. We also use 37 of them in this study. Authors in [7] did not divide applications into categories like Streaming, Email, etc. Nevertheless, they considered the importance of application classification for network security and trend analysis, and divided applications into popular end-user applications such as Facebook, Skype, etc. The idea of categorizing end-user applications is applied in this paper.

The problem of ML algorithm is large training time which makes it ineffective of real time traffic classification. Solution of this problem is to reduce the number of features that represent the application type. The work of paper [8] shows that the feature selection algorithm can reduce the training time of the Bayes Net algorithm, making the Bayes Net classifier more suitable for real-time and online IP traffic classification. Williams [9] certified that feature selection could improve computational performance without sacrificing classification accuracy. In our study, we classify popular end-user applications such as iQiyi, iQiyiVR, Youku, YoukuVR and non-video.

2.2 Brief Introduction of ML

In this paper, we evaluate video application identification performance with six commonly used ML algorithms. Next, we briefly introduce the basic concepts of these six algorithms.

(1) **Naïve Bayes** classifiers are a family of simple "probabilistic classifiers" based on Bayes theorem. Naïve Bayes assumes strong independence between features [8]. The probability that an instance x belongs to a class c can be expressed as:

$$P(C = c|X = x) = \frac{P(C = c)\prod_{i} P(X_i = x_i|C = c)}{P(X = x)} \tag{1}$$

Where X is a vector of instances where each instance is described by features $\{X_1, X_2, \cdots X_k\}$, and C is the class of an instance [9].
We evaluate Naïve Bayes with discretization (NBD) which converts successive features into discrete features in this paper.

(2) **Bayesian Network** is a directed acyclic graph model [10]. The nodes of the model represent features or classes, and the links between nodes represent their probabilistic relationship.

(3) **K-Nearest Neighbors (KNN)** calculates the Euclidean distance from each test instance to the k nearest neighbors [11]. The k nearest neighbors vote to determine the class of test instance.

(4) **AdaBoost** is a meta-learning algorithm, which is built from a linear combination of simple classifiers. AdaBoost uses several classification models to decide the class label of an instance [12].

(5) **C4.5 Decision Tree** is a tree structure (a binary tree or a non-binary tree). Each non-leaf node indicates a test on features. Each branch indicates the output of the feature in a range of values, and each leaf node stores a category. In order to determine the class of a test instance, C4.5 Decision Tree starts testing the feature attributes corresponding to the test instance from the root node. Then this algorithm selects the output branch according to the value of the feature attribute. C4.5 Decision Tree repeats this process until it reaches the leaf node [13]. The category stored in the leaf node is the decision result.

(6) **Random Forest** is a classifier that contains multiple classification trees. All trees in the forest have the same distribution. The output category is determined by the mode of the individual tree's output [14].

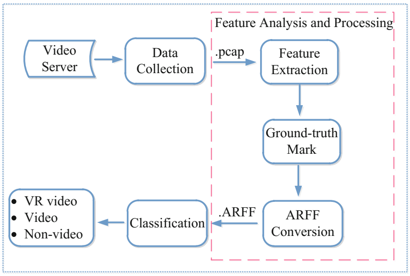


Fig. 1. The outline of identification system

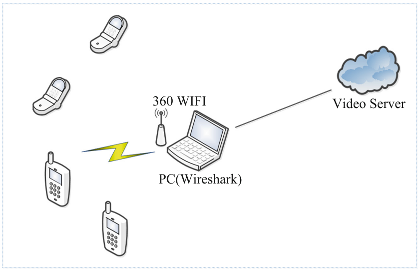


Fig. 2. The data collection system for dataset gathering

3 VR Video Application Identification System

We present the outline of our identification system in Fig. 1, which consists of three parts: (1) Data collection, (2) Feature analysis and processing module for flow feature extraction, ground-truth mark, and ARFF conversion, (3) Classification module. The definition of a flow based on 5-tuple (source IP address, destination IP address, source port, destination port, and protocol) is adopted in this paper.

3.1 Data Collection

Our dataset is gathered via Wireshark [15]. Traffic is collected with five categories, e.g., iQiyiVideo, iQiyiVRVideo, YoukuVideo, YoukuVRVideo and non-video. Among these categories, a total of seven applications are applied. More details are given in Table 1.

As most VR video applications, such as 3D broadcast, iQiyiVR, and YoukuVR, are running on mobile devices, we collect mobile traffic with smartphones and iPad via WIFI access. At the same time, the computer runs Wireshark to collect traffic, which is stored in.pcap format for subsequent processing. The architecture of the data collection system is depicted in Fig. 2.

3.2 Feature Analysis and Processing

Appropriate flow feature acquisition is the premise of using ML algorithms to classify network traffic. In this section, we firstly analyze the different statistical features between VR video and ordinary video in detail. Then we introduce the further processing of these features.

Table 1. Categories, applications and the number of instances in our dataset

Category	Application	Number	Percentage of total (%)
iQiyiVideo	iQiyi	6993	36.74
iQiyiVRVideo	iQiyiVR	3800	19.96
YoukuVideo	Youku	2669	14.02
YoukuVRVideo	YoukuVR	1494	7.85
Non-video	Zhihu, Mail, Taobao	4078	21.43

Flow Feature Analysis. In this part, the current mainstream VR video applications such as Storm Mirrors VR, 3D broadcast, and Youku VR, etc. are investigated. The main differences between VR video service and most traditional video services are the process of increasing multi-camera video splicing and 360° video projection before video coding. Nevertheless, almost all current VR videos are still encoded in H264, which is the same as ordinary video. Therefore, the main coding related parameters for VR video are still resolution, bit rate, frame rate, etc.

However, due to the characteristics of VR video, higher requirements are placed on these video parameters. As seen from Fig. 3, the Field of View (FOV) in VR video is only part of the entire video. In order to achieve the appropriate resolution for the FOV, the entire VR video requires very high resolution. Take a 4 K (3840 $\times$ 1920) VR video as an example. Assuming that the HMD's angle of view is 90° in both directions of the horizontal α and the vertical β, the video resolution in the FOV is only 960 $\times$ 480, which is far away from the near-future 4 K video requirement. In order to improve users' experience, VR video requires even higher resolution.

High-resolution video requires a higher bit rate. In order to save the packet packaging cost, each packet size of VR video will be larger than that of ordinary video. Therefore, the average packet throughput and average byte throughput of VR video will be larger than that of ordinary video. We exploit traffic capture and analysis to obtain the differences of flow features between VR video and ordinary video.

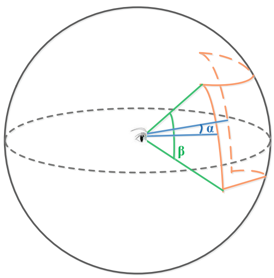


Fig. 3. VR video spherical projection and the FOV

We capture traffic and save them in.pcap format files, namely youkuvr.pcap and youku.pcap, for VR video and ordinary video, respectively. Each.pcap file has about 180,000 packets, which is equal to 124 MB. However, it takes 325 s to capture VR video and 770 s to capture ordinary video. Thus, VR video is two times the average packet throughput and byte throughput of the ordinary video. On the other hand, we analyze the difference of packet arrival interval between ordinary video and VR video. Packet arrival interval of them are given in Fig. 4a, b, respectively. There is a big difference between them. In terms of the packet arrival interval of ordinary video, the overall trend is relatively flat, and there are some protrusions in short time (about 30 s). However, for VR video, there are some large protrusions in a relatively long period of time (about 40 s). The maximum number of packets arriving per second is also dif-

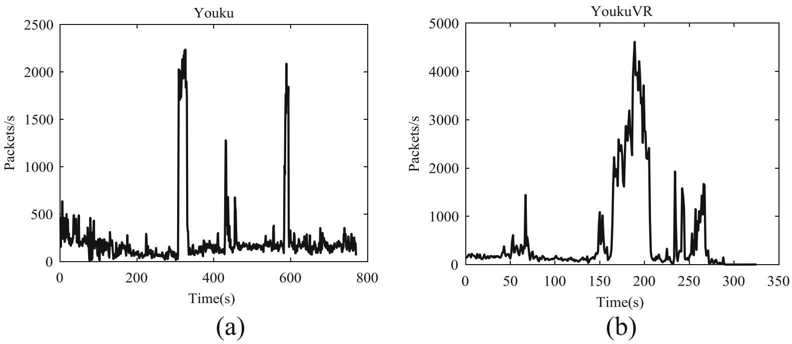


Fig. 4. **a**. Packet arrival interval of ordinary video **b**. Packet arrival interval of VR video

Table 2. Statistical features in this paper

	Before feature selection (37)	After feature selection (22)
Features	# Protocol, source and destination ports	# Protocol, source and destination ports
	# The number of packets/bytes	# The number of packets/bytes
	# The number of packets without Layer 4 payload	# The number of packets without Layer 4 payload
	# Start time, end time, duration	# Start time, end time, duration
	# Average packet throughput, average byte throughput	# Average packet throughput, average byte throughput
	# Max/min/average/standard deviation of packet sizes and inter-arrival times	# Max/min/average/standard deviation of packet sizes and inter-arrival times
	# Number of TCP packets with FIN, SYN, RSTS, PUSH, ACK, URG, CWE, ECE flags set (all zero for UDP packets)	# Number of TCP packets with FIN, SYN, RSTS(all zero for UDP packets).
	# The size of the first ten packets.	

ferent, which are 2200 and 4500, respectively. We also find that the packet sizes of them are different. In summary, the average packet/byte throughput, packet arrival interval and packet size will be representative features. Initially, we select 37 unidirectional flow features from those in [3] according to the findings made in above data analysis. The 37 features are shown in Table 2 (the column on the left).

Ground-truth Mark. In our study, each traffic from the video/VR video application is labeled as video or VR video. So, the classifiers will model some non-video features as video features, such as the features of ads and others. Yet the goal in this study is to ensure the overall experience of people using video applications.

ARFF Format Conversion. We make use of the PostgreSQL database to store data so that we can conveniently convert.pcap format into ARFF format. In the ARFF file, each flow is considered as an instance. The number of instances for each category is shown in Table 1.

4 Experimental Results

Through extensive experiments, we try to observe: (a) the identification performance for VR video application using the proposed statistical features; (b) the best algorithm for VR video application identification, considering both accuracy and build time; (c) the effect of feature selection on algorithm performance; (d) the change of overall accuracy in respect to the size of the training set.

The classification module is mainly composed of six most often-used supervised ML algorithms from WEKA [16]: Naïve Bayes, Bayesian Network, KNN (k is chosen as 1 in our experimental setup), AdaBoost (J48 is the Base classifier), J48 (C4.5 Decision Tree in WEKA), and Random Forest. The ML algorithms applied in this study are all implemented using the WEKA tool and only a few parameters are adjusted. Our test option is set to 10-fold cross validation, by which we gain the best overall accuracy during the entire experiments. We test 5-fold, 10-fold, 15-fold and 20-fold in our experiments.

4.1 Performance Metrics

To evaluate the performance of the six ML algorithms using the proposed statistical features, we use three metrics: overall accuracy (Acc), F-measure (F1) and build time.

First, we introduce the definition of True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP means that the forecast is positive and actually positive. FP means that the forecast is positive but actually negative. TN means that the forecast is negative and actually negative. FN means that the forecast is negative but actually positive.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$F1 = \frac{2 * Precision * Recall}{\Pr ecision + \operatorname{Re} call} \tag{3}$$

where Precision is $\frac{TP}{TP+FP}$, and Recall is $\frac{TP}{TP+FN}$.

Acc is applied to measure the accuracy of an algorithm on the whole dataset. F1 is to evaluate the identification performance for each category. Build time is the time taken to create an identification model given a training set.

4.2 Classification Results Using 37 Features

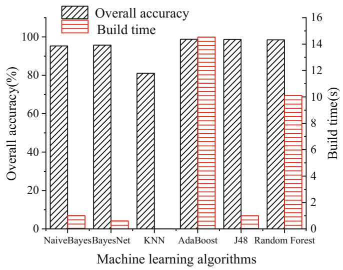


Fig. 5. Overall accuracy and the build time using 37 features

The results of our dataset are given in Fig. 5. The Acc and the build time of each algorithm are shown in Table 3. The evaluation criteria for each strategy is F1, given in Sect. 4.1.

The results show an overall accuracy of 81% to 98.7%. In addition to KNN algorithm, the accuracy of other algorithms is above 95%. This phenomenon indicates that we can effectively distinguish VR video from ordinary video using the proposed statistical features. At the same time, it indicates that the proposed statistical features have good performance on various ML algorithms, and they are universal. These statistical features can also be used to distinguish video from non-video.

As shown in Fig. 5, AdaBoost algorithm gives the best accuracy and the longest build time. In addition, the accuracy of J48 algorithm is similar to that of AdaBoost algorithm, but J48 algorithm builds model in a shorter period of time, only one second. This is explained by the fact that J48 algorithm has less training demands, and that it has lower complexity than AdaBoost algorithm.

In general, the proposed features combined with J48 algorithm can achieve good performance for VR video application identification. They can achieve an overall accuracy of 98.6%, and the build time is about one second.

4.3 Further Discussions

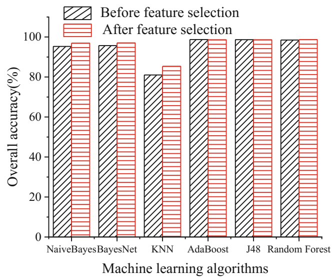
Considering the requirement of real-time network traffic identification, we make two improvements to the experiments, i.e., study feature selection and the change of the overall accuracy in respect to training set size.

Feature Selection. We adopt the Principal Components Analysis (PCA), one of the most well-known algorithms in feature selection. This algorithm looks for a series of

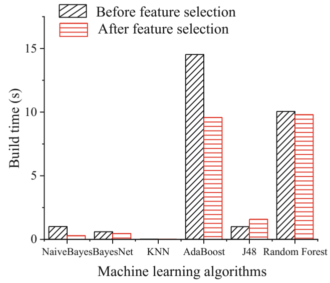
Table 3. The accuracy (%) and the build time(s) of each category

	NaiveBayes	BayesNet	KNN	AdaBoost	J48	Random Forest
iQiyi	95.7	96.0	87.3	98.9	98.8	98.6
iQiyiVR	95.9	96.4	81.1	98.9	99.0	99.1
Youku	93.4	93.7	75.6	97.9	97.7	97.4
YoukuVR	93.2	94.2	63.0	98.1	98.2	97.7
Non-video	95.7	96.2	79.5	98.9	98.9	98.9
Acc	95.3	95.7	81.0	98.7	98.6	98.5
Time	1.01	0.6	0.01	14.53	1	10.1

projection directions. After the high-dimensional data are projected in these directions, the variance is maximized. The first principal component is the largest variance, and the second principal component is the second largest variance. PCA algorithm selects the first 22 features of the 37 features, as shown in Table 2 (the column on the right). The overall accuracy and the build time with the 22 features and all features are given in Fig. 6a, b, respectively. After feature selection, the overall accuracy is hardly changed and even some algorithms have slight accuracy improvement. In addition, the build time is greatly shortened so that it is qualitatively consistent with the result in [9].



(a)



(b)

Fig. 6. **a**. Overall accuracy of six algorithms using all features and selected features **b**. The build time of six algorithms using all features and selected features

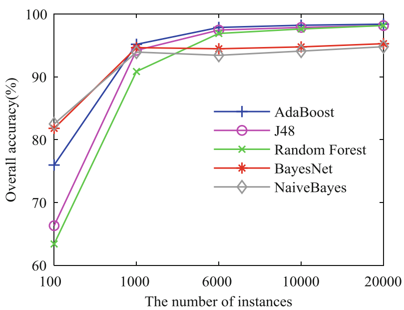


Fig. 7. The change of the overall accuracy in respect to training set size

Therefore, the 22 features in Table 2 (the column on the right) are more suitable for VR video identification.

The Change of the Overall Accuracy in Respect to Training Set Size. We consider 19,034 network flows in our study, which are too large in real-time traffic identification. Therefore, we study the change of the overall accuracy in respect to training set size. The details are given in Fig. 7. Here we ignore KNN algorithm. We can find that when the number of flows exceeds 1000, AdaBoost can always provide the best performance, followed by J48 (and it is quite fast to train) and Random Forest. Due to the scarcity of training data in real-time network traffic identification, it is very exciting that 6000 network flows can provide good identification results.

5 Conclusions

In this study, we proposed 22 statistical features that can well represent VR video application. Classification strategies such as iQiyiVideo, iQiyiVRVideo, Youku-Video, YoukuVRVideo and non-video, etc. were adopted, and we evaluated and obtained the C4.5 Decision Tree algorithm which performed the best in terms of overall accuracy and the build time. These 22 main statistical features combined with C4.5 Decision Tree algorithm could achieve an accuracy of 98.6% for VR video application identification while maintaining high computational performance. In addition, this paper proved that as long as the training set exceeds 6,000 flows, high accuracy could be achieved, which makes it possible to identify real-time video application. Our work can effectively distinguish VR video from ordinary video, which provides a good foundation for other works such as resource scheduling.

Acknowledgements. This work has been sponsored by Huawei Research Fund (grant No. YBN2016110032) and National Science Foundation of China (No. 61201149). The authors would also like to thank the reviewers for their constructive comments.

References

1. IANA, http://www.iana.org/assignments/port-numbers. Accessed 06 June 2018
2. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC: multilevel traffic classification in the dark. In: ACM SIGCOMM Computer Communication Review, Pennsylvania, pp. 229–240 (2005)
3. Kim, H., Claffy, K.C., Fomenkov, M., Barman, D., Faloutsos, M., Lee, K.: Internet traffic classification demystified: myths, caveats, and the best practices. In: Proceedings of the 2008 ACM CoNEXT Conference, Spain, p. 11 (2008)
4. Andersson, R.: Classification of video traffic: an evaluation of video traffic classification using random forests and gradient boosted trees. Digitala Vetenskapliga Arkivet. **83** (2017)
5. Västlund, F.: Video flow classification: a runtime performance study. Digitala Vetenskapliga Arkivet. **68** (2017)
6. Moore, A., Zuev, D.,Crogan, M.: Discriminators for use in flow-based classification. (2013)

7. Yamansavascilar, B., Guvensan, M. A., Yavuz, A. G., Karsligil, M. E.: Application identification via network traffic classification. In: International Conference on ICNC, Santa Clara, pp. 843–848(2017)
8. Aggarwal, R., Singh, N.: A new hybrid approach for network traffic classification using SVM and Naïve Bayes algorithm. Int. J. Comput. Sci. Mobile Comput. **6**, 168–174 (2017)
9. 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), 5–16 (2006)
10. Williams, N., Zander, S.: Evaluating machine learning algorithms for automated network application identification. (2006)
11. Chen, Z., Chen, R., Zhang, Y., Zhang, J., Xu, J.: A Statistical-Feature ML Approach to IP Traffic Classification Based on CUDA. In: IEEE Trustcom/BigDataSE/ISPA, Tianjin, pp. 2235–2239 (2017)
12. Datta, J., Kataria, N., Hubballi, N.: Network traffic classification in encrypted environment: a case study of google hangout. In: Twenty First National Conference on Communications (NCC), pp. 1–6 Mumbai, India (2015)
13. Munther, A., Alalousi, A., Nizam, S., Othman, R. R., Anbar, M.: Network traffic classification-A comparative study of two common decision tree methods: C4.5 and Random forest. In: International Conference on Electronic Design, pp. 210–214 Penang (2014)
14. Wang, C., Xu, T., Qin, X.: Network traffic classification with improved random forest. In: International Conference on Computational Intelligence and Security, pp. 78–81 Shenzhen (2015)
15. Wireshark, https://www.wireshark.org/. Accessed 22 May 2018
16. WEKA: Data Mining Software in Java. https://www.cs.waikato.ac.nz/ml/weka. Accessed 26 May 2018