ChinaRxiv

AI translation · View original & related papers at
chinarxiv.org/items/chinaxiv-201808.00059

P2P traffic identification method based on clustering and traffic propagation graph (Postprint)

Authors: Yangyang Su, Sun Dongpu, Li Dandan, Sun Guanglu

Date: 2018-08-13T00:00:00+00:00

Abstract

To effectively regulate networks and rapidly, accurately identify P2P traffic, this work proposes a P2P traffic identification method based on network behavioral characteristics. By analyzing interaction and behavioral features between nodes and between nodes and links in P2P network traffic, the approach integrates clustering methods with traffic propagation graph techniques. The method first clusters traffic from different network application types by collecting packet-level and flow-level statistical features of network flows, then utilizes traffic propagation graphs to identify P2P traffic. Experimental results demonstrate that the proposed method effectively identifies P2P network application traffic on backbone network data, achieving an F1-measure exceeding 95%.

Full Text

Preamble

P2P Traffic Identification Method Based on Clustering and Traffic Dispersion Graph

Su Yangyang[1], Sun Dongpu[1], Li Dandan[1,2], Sun Guanglu[1,2]
(1. School of Computer Science & Technology; 2. Research Center of Information Security & Intelligent Technology, Harbin University of Science & Technology, Harbin 150080, China)

Abstract: To effectively supervise networks and rapidly and accurately identify peer-to-peer (P2P) traffic, this paper proposes a P2P traffic identification method based on network behavioral characteristics by analyzing the interaction and behavioral features between nodes and between nodes and links in P2P network traffic, and by combining clustering methods with traffic dispersion graph techniques. The method first clusters traffic from different network applications based on collected packet-level and flow-level statistical features, then identifies P2P traffic using traffic dispersion graphs. Experimental results

demonstrate that the proposed method can effectively identify P2P network application traffic on backbone network data, achieving an F1-measure of over 95%.

Keywords: P2P traffic identification; traffic behavioral characteristics; traffic dispersion graph; density-based spatial clustering of applications with noise

0 Introduction

Peer-to-peer (P2P) networks are distributed network models that operate without intermediate entities. In recent years, with the rapid development of computer networks, many network applications have adopted P2P technology principles to implement their services. Consequently, P2P protocols have been widely applied in instant messaging, video sharing, file sharing, online live streaming, gaming, and other domains. According to Cisco' s annual traffic statistics report, although the proportion of P2P traffic occupying global network bandwidth shows a declining trend, it still reaches 40% of total bandwidth capacity [1]. Since P2P applications employ multi-connection patterns to ensure data transmission efficiency, they consume substantial network bandwidth and can easily trigger network congestion issues. Therefore, accurately identifying P2P traffic within overall network traffic and effectively supervising it holds significant importance.

Existing P2P traffic identification methods primarily rely on analyzing the unique characteristics of P2P networks and traffic to discover their distinctive static and dynamic features, thereby effectively distinguishing P2P traffic from other network traffic to help network administrators and service providers improve quality of service for different network applications. Current P2P traffic identification methods mainly include port-based identification, payload-based identification, statistical feature and machine learning-based identification, and network node relationship and host behavior-based identification methods [2]. These approaches analyze and identify P2P traffic from different perspectives, each with its own advantages and disadvantages.

Since most existing P2P applications use dynamic ports and encryption for transmission, port-based and payload-based methods cannot effectively identify them [3,4]. While statistical feature and machine learning-based identification methods do not solely depend on ports and payload, the unstable value ranges of flow statistical features across different network environments create large discrepancies between training and test data, thereby affecting the identification effectiveness of supervised machine learning models. Moreover, these methods exhibit poor adaptability to newly emerging protocols in different network environments [5,6].

Although network node relationship and host behavior-based methods can identify new protocols, they are limited by changes in network topology environ-

ments and are difficult to apply in high-speed backbone networks. Iliofotou et al. [7] proposed the concept of Traffic Dispersion Graphs (TDG), which converts communication relationships between nodes into directed graphs to mine deep network interaction behaviors. They quantify features such as in-degree, out-degree, network diameter, and maximum connected components in directed graphs, using these features to identify application types of communication links. However, not all communication links in networks are mutually connected, and even mutually connected links may not belong to the same network application simultaneously. Therefore, using TDG alone for different traffic identification may misclassify small flows with insufficiently obvious feature attributes or even fail to identify them.

Consequently, this paper proposes CTDG (Clustering and Traffic Dispersion Graph based method), an improved P2P traffic identification method combining clustering with TDG graph models. The CTDG method first clusters network flows with similar statistical features collected from network traffic into several potentially identifiable classes using an unsupervised machine learning model, then utilizes metrics defined in TDG graphs to quantify interaction behavioral characteristics among network flows for P2P traffic identification. Experimental results demonstrate that the proposed method achieves significant effectiveness in identifying P2P traffic in high-speed backbone networks, with accuracy reaching over 95%.

1 Related Work and Existing Problems

Among existing P2P traffic identification methods, statistical feature and machine learning-based approaches do not rely on application layer payload content. Instead, they analyze and extract traffic statistical features based on network and transport layers, combine them with labeled traffic datasets, and train models in supervised machine learning frameworks to ultimately identify traffic from various applications. These methods typically utilize packet-level features and flow-level features. Packet-level features mainly include port numbers, average packet arrival time, maximum Ethernet packet size, and maximum inter-packet time intervals. Flow-level features primarily include duration of individual flows, flow length, and inter-flow intervals.

Xu et al. [8] identified traffic by constructing a dynamic hybrid identification strategy combining SVM with a voting mechanism. Roughan et al. [9] proposed nearest neighbor and linear discriminant analysis methods based on the aforementioned statistical features. Liu et al. [10] proposed 26 statistical features for P2P flows and used support vector machine models to distinguish four types of P2P traffic, achieving favorable identification results, though they struggled to effectively identify application categories with few flows. Sun et al. [11] proposed a P2P game traffic identification method based on flow characteristic description, analyzing packet distribution of labeled critical traffic, using membership functions as evaluation sets, and finally applying fuzzy evaluation criteria to determine network applications for P2P traffic identification. However, this

method heavily relies on payload data and exhibits poor adaptability for P2P network applications with inconspicuous identification features and encryption. Chen [12] conducted research on early P2P traffic identification based on SVM, using early packets in data flows for feature selection and identification. Dai et al. [13] employed active learning techniques to extract a small number of high-quality samples and used support vector machine modeling for P2P traffic identification, though applying this to actual complex network environments requires case-by-case analysis.

Network traffic classification methods based on network node relationships and host behavior focus on the roles hosts play in networks, connection patterns between hosts, and certain group behaviors in networks. Karagiannis et al. [14] pioneered the use of peer connection patterns in P2P networks to identify P2P traffic in networks [12], subsequently proposing a network traffic classification method based on host behavior patterns (BLINC). The BLINC method categorizes host behavior patterns into social, functional, and application layers, identifying network traffic by extracting these behavior patterns. While this method improved P2P traffic matching accuracy, it overly depended on relationships between ports and IPs. Hu [15] proposed a traffic identification method combining hybrid behavioral features with the Spark big data parallel framework. Constantinou et al. [16] obtained P2P network connection topology graphs based on actual connection establishment between each node and other nodes, discovering that P2P network topology graphs have larger network diameters compared to other network types, thereby making the method' s data processing and metric computation system requirements high and difficult to achieve convenient usability. Lu et al. [17] proposed a P2P node identification algorithm based on node connection characteristics, utilizing the number of connections between nodes and destination subnets per unit time and the ratio of connections to effective connections. Although this algorithm' s processing time is shorter than deep packet inspection, it relies more on transport layer features for P2P traffic identification.

To address the aforementioned limitations, this paper proposes an improved P2P traffic identification method combining clustering and TDG graph models (Clustering and Traffic Dispersion Graph based method, CTDG). This method offers the following advantages: (a) it does not require payload content and can identify encrypted P2P traffic; (b) it mines deep P2P network interaction behaviors to distinguish graph characteristics from other application networks for effective P2P application identification; (c) it exhibits good applicability for newly emerging network applications without requiring training or complex model parameter configuration.

2 Network Traffic Statistical Feature Extraction

This paper defines network flows using the commonly employed five-tuple information (source IP, destination IP, source port, destination port, transport layer protocol) and uses bidirectional flows within a certain time period as the

basic unit. For TCP flows, the direction is defined with the sender of the first packet as the source and the receiver as the destination. For UDP flows, the direction is similarly defined with the sender of the first packet with identical five-tuple information as the source and the receiver as the destination. This paper extracts statistical features of network flows, network node relationships, and host behavior features.

Network flows generated by different application layer protocols exhibit significant differences in packet-level and flow-level features [18]. This paper extracts 60 network flow statistical features including flow size, session duration, packet arrival times within flows, number of bidirectional packets, inter-packet arrival time intervals (mean, variance), and idle time spent by communicating parties. The information gain algorithm [19] is used to select the most relevant features as clustering attributes, as shown in Table 1 .

Table 1 Flow Statistical Features

Packet-level Features	Flow-level Features
Byte lengths of first 6 packets Maximum, minimum packet length Packet length mean, variance	Flow duration Flow arrival interval time {Source IP, destination IP, source port, destination port, transport layer protocol}

3 Traffic Dispersion Graph (TDG)

This paper defines the TDG graph among all network nodes using a directed graph $G(V, E)$, where the node set V represents network nodes, and edges $(u, v) \in E$ represent network flows from host u to host v.

In P2P networks, each node can independently determine its own communication behavior, yet nodes exhibit interdependence through link communication for information and resource sharing. The TDG graph of P2P network nodes (Figure 1 [Figure 1: see original paper]) possesses the following characteristics:

a) **High average node degree**. This results from numerous P2P nodes interconnecting to achieve data sharing and content querying.

b) **Large proportion of nodes with both in-degree and out-degree**. This reflects the dual server-client identity characteristic of numerous P2P nodes in the network.

c) **Large network diameter for some P2P networks**. This stems from the decentralized network topology structure of P2P applications such as BitTorrent.

4 DBSCAN Clustering Algorithm

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a typical density clustering algorithm. Its core concept derives the maximum density-connected sample set through density-reachable relationships as the final category. It describes the compactness of sample sets using neighborhoods, with parameters eps and minPTS describing the distribution density of neighborhood samples. Parameter eps defines the neighborhood distance threshold for a sample, while minPTS defines the sample count threshold within the eps distance neighborhood. The basic algorithm flow is as follows:

a) DBSCAN obtains all core objects from samples, i.e., for each sample, based on the distance metric, it identifies samples satisfying the eps neighborhood distance and greater than minPTS as core object members.

b) In the core object set, it randomly selects an object, initializes the current cluster core object queue, class ID, current cluster sample set, and unvisited sample set. By iteratively selecting each object in the current cluster core object queue, it finds the neighborhood subset within the eps distance threshold, updates the current cluster sample set and unvisited sample set, and simultaneously adds core object samples from the neighborhood subset to the current cluster core object queue.

c) If the current cluster core object queue no longer increases, the current cluster becomes a new category C_k and is added to the cluster partition set $C = \{C_1, C_2, ..., C_K\}$. This process continues until every member in the core objects is partitioned into the cluster partition set, concluding the clustering.

Since the DBSCAN algorithm can identify clusters of different shapes and demonstrates strong robustness to noise points [20], this paper employs it for network flow shunting processing before traffic identification.

Different distance calculation methods directly affect DBSCAN clustering effectiveness. Traditional DBSCAN uses Euclidean distance, which focuses more on absolute distances between feature values and often neglects relative distances between samples. For P2P network data flows, relative distance comparison more accurately characterizes the relative relationships between samples. Therefore, this paper proposes using chi-square distance in DBSCAN to measure relative distances between samples. The chi-square distance formula is:

$$d(x, y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{x_i + y_i}$$

Chi-square distance, derived from chi-square statistics, has been widely applied in practical distance measurement problems with considerable success [21].

5 CTDG Traffic Identification Method

Based on TDG graph definitions and P2P network graph model characteristic analysis, this paper constructs the following behavioral features:

a) **Percentage of nodes with both in-degree and out-degree among all nodes**, denoted as IO.

b) **Network diameter**: the distance between the two nodes with the longest shortest path among all node pairs.

c) **Average node degree**.

The CTDG method combines improved clustering with TDG graph relationship mining, using network traffic statistical features and behavioral features to achieve effective P2P traffic identification. The CTDG identification process is shown in Figure 2 [Figure 2: see original paper] and consists of four steps:

a) Filtering. Since port-based and payload-based methods effectively identify certain non-encrypted traditional applications, this paper applies these methods to filter out recognizable applications such as Web, DNS, and SMTP. This not only reduces interference from other background traffic but also decreases time and space complexity in subsequent steps.

b) Shunting. Using the statistical features listed in Table 1, DBSCAN clustering groups network flows with similar statistical features into clusters. The algorithm employs Euclidean distance to calculate similarity in feature space. Let F represent the network flow dataset, and $f_i \in F$ represent each network flow. The detailed algorithm steps are:

1. For each unprocessed flow f_i in the dataset, scan its radius (eps) and detect flows within the eps coverage range. If the count exceeds the minimum flow threshold (minPTS), create a new cluster Y and add these flows to cluster Y.
2. For each flow f_j in cluster Y, detect flows within its eps coverage range. If the count is greater than or equal to minPTS, add flows not contained in any cluster to cluster Y.
3. Repeat step (2) until no new network flows are added to cluster Y.
4. Based on identification results, repeat steps (1)-(3) until all network flows are processed.

c) Merging similar clusters. IP similarity is defined as the ratio of identical IPs appearing in two clusters to the total number of IPs in both clusters. If the IP similarity fails to meet the predefined threshold, the merging process terminates.

Ideal clustering results would group all flows from the same application into one cluster containing only that application' s traffic. However, practical clustering reveals that the same application can generate multiple clusters. Analysis

shows that P2P protocols have multiple interaction modes: they generally use UDP for query processes and TCP for file transfers, with these communication modes exhibiting significant differences in packet-level and flow-level statistical features. Since different clusters generated by the same application correspond to TDGs with numerous common nodes, this paper uses IP similarity as the cluster merging condition.

d) Classification using TDG metrics. After merging, each flow group creates a TDG and uses its metric indicators for classification. TDG metrics include: the percentage of nodes with both in-degree and out-degree among all nodes, network diameter constraints, and average node degree as TDG classification indicators.

TDGs are created from the different clusters obtained above, and their metric values are calculated. If the metric values satisfy the set thresholds, the TDG is determined to match the P2P pattern, and each flow within it is labeled as P2P application.

6 Experiments

6.1 Dataset

This paper uses traffic collected at different times in 2017 from a Chinese backbone network as the experimental dataset. Table 2 provides detailed dataset descriptions. The method extracts network flow statistical features and uses CoralReef to process network traffic. CoralReef is a software suite for passive analysis of Internet traffic. A flow timeout value of 64 seconds is configured, and payload-based feature matching is used to label the dataset.

Table 2 Traffic Dataset Information

	Backbone1	Backbone2
Traffic duration	5 min	30 min
Number of packets		
Packet bytes	80 GB	

Through manual labeling and analysis, the experimental dataset primarily includes DNS, Web, P2P, Streaming, Games, Network-operation, MAIL/NEWS, and other network application protocol types, plus some applications unrecognizable by payload analysis methods. During experiments, flows that were difficult to identify and flows without payload were removed. Figure 3 [Figure 3: see original paper] describes the distribution of application types in the two network traffic datasets.

6.2 Evaluation Method

To accurately evaluate the proposed method, precision, recall, and comprehensive evaluation index (F1-measure) are adopted. The metrics are defined as:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\text{-}measure = \frac{2PR}{P + R}$$

Where: TP (true positives) represents the number of samples correctly classified as P2P; FP (false positives) represents the number of non-P2P samples incorrectly identified as P2P; and FN (false negatives) represents the number of P2P samples incorrectly identified as non-P2P.

6.3 Experimental Results and Analysis

First, the effectiveness of the DBSCAN algorithm in clustering flows belonging to the same application is tested. Based on the benchmark method' s labeled flow types, each cluster is labeled with the application type containing the most labeled flows, i.e., all flows in the cluster are marked as that application type. The DBSCAN algorithm adjusts the final number of clusters and clustering results by tuning parameters eps and minPTS. Smaller minPTS values generate more clusters. After determining the minimum minPTS, classification performance continuously improves with increasing eps. However, when eps becomes too large, classification performance significantly degrades. As shown in Figure 4 [Figure 4: see original paper], the algorithm achieves optimal performance when eps is 0.02-0.04 and minPTS=4, with cluster labeling accuracy exceeding 90%.

Step c) in the CTDG method merges multiple clusters that may be running the same application. The merging effectiveness depends on the node similarity threshold setting. An excessively large threshold makes cluster merging difficult, causing network flows from the same application to distribute across different clusters, which hinders comprehensive analysis of behavioral patterns for the same application type. Conversely, an overly small threshold incorrectly merges clusters from different applications, reducing overall identification accuracy.

As shown in Figure 5 [Figure 5: see original paper], when the node similarity threshold is set at 0.4-0.7, the CTDG classification method achieves over 90% accuracy, indicating good classification performance. With clustering parameters minPTS=4, eps=0.025, and node similarity threshold of 0.6, the CTDG method achieves 93% recall and 96% accuracy. At eps=0.03, CTDG also achieves over

90% recall and accuracy. However, experiments also revealed that improper parameter selection can significantly degrade CTDG classification performance.

Compared with the CTDG method, BLINC identifies each host' s flows based on transport layer connection patterns (such as port and IP relationships). When applied to the current dataset, BLINC achieves 84% accuracy and 89% recall. Additionally, BLINC exhibits low identification rates for some P2P applications like BitTorrent, reaching only 25%, whereas the CTDG method achieves 90% detection rates, as shown in Table 3 . Since CTDG introduces a clustering process and utilizes more statistical features as clustering metrics, TDG construction is more effective, significantly improving final identification performance. Compared with Chen' s SVM_{PF} method for P2P traffic identification, which uses early bidirectional flow packets as feature selection basis, CTDG achieves higher recall rates (generally below 85% for SVM_{PF}), as shown in Figures 6 [Figure 6: see original paper] and 7 [Figure 7: see original paper].

Table 3 Performance Comparison of CTDG, SVM_{PF}, and BLINC Methods (%)

Method	Precision	Recall	F1-measure
BLINC			
SVM_{PF}			
CTDG			

7 Conclusion

This paper addresses the network behavioral characteristics of P2P traffic, applying a clustering method based on packet-level and flow-level statistical features of network flows combined with TDG graph features. The proposed CTDG method integrates network flow and host behavior features with TDG for more accurate and effective P2P traffic identification. Experiments demonstrate that this method achieves notable improvements in precision, recall, and F1-measure compared with BLINC and SVM_{PF} methods. The contribution of this paper lies in providing a new research approach for solving traditional P2P traffic identification problems.

References

[1] Cisco Systems. Cisco visual networking index: forecast and methodology, 2010-2015 [R]. 2011.

[2] Kim H, Claffy K, Fomenkov M, et al. Internet traffic classification demystified: myths, caveats, and the best practices [C]// Proc of ACM CoNEXT Conference. [S. l.]: ACM Press, 2008: 1-12.

[3] Sun Boen. Research on detection method of Peer-to-peer botnet in high-speed network environment [D]. Chengdu: University of Electronic Science and

Technology, 2016.

[4] Niu Weiba, Zhang Xiaosong, Sun Boen, et al. Two-stage peer-to-peer zombie network detection method based on flow similarity [J]. University of Electronic Science and Technology, 2017, 46(6): 902-906, 948.

[5] Wang Chunzhi, Du Yuanli, Ye Zhiwei. Peer-to-peer traffic recognition based on optimal ABC-SVM algorithm [J]. Application Research of Computers, 2018, 35(2): 582-585.

[6] Dainotti A, Pescape A, Claffy K. Issues and future directions in traffic classification [J]. IEEE Network, 2012, 26(1): 35-40.

[7] Iliofotou M, Pappu P, Faloutsos M, et al. Network monitoring using traffic dispersion graphs [C]// Proc of ACM SIGCOMM conference on Internet measurement. [S. l.]: ACM Press, 2007: 315-320.

[8] Xu Hongping, Liu Yang, Yi Hang, et al. Anomaly flow identification technology for launch vehicle detection network [J]. Journal of Tsinghua University: Natural Science, 2018, 58(1): 20-26, 34.

[9] Roughan M, Sen S, Spatscheck O, et al. Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification [C]// Proc of ACM SIGCOMM conference on Internet measurement. [S. l.]: ACM Press, 2004: 135-148.

[10] Liu H, Feng W, Huang Y, et al. A peer-to-peer traffic identification method using machine learning [C]// Proc of International Conference on Networking, Architecture and Storage. 2007.

[11] Sun Zhixin, Gong Jing. A method of fuzzy recognition of peer-to-peer traffic based on flow characteristic description [J]. Chinese Journal of Computers, 2008, 31(7): 1252-1260.

[12] Chen Yang. Research on the early recognition of Peer-to-peer traffic based on SVM [D]. Baoding: Hebei University, 2017.

[13] Dai Lei, Yun Xiaochun, Zhang Yongzheng, et al. A peer-to-peer flow recognition technique based on TCM active learning [J]. Hi-Tech Newsletter, 2010(7): 23-29.

[14] Karagiannis T, Papagiannaki K, Faloutsos M. BLINC: multilevel traffic classification in the dark [J]. ACM SIGCOMM Computer Communication Review, 2005, 35(4): 229-240.

[15] Hu Bin. Research and application of traffic recognition technology based on mixed behavior characteristics [D]. Beijing: Beijing University of Posts and Telecommunications, 2017.

[16] Constantinou F, Mavrommatis P. Identifying known and unknown peer-to-peer traffic [C]// Proc of IEEE International Symposium on Network Computing and Applications. [S. l.]: IEEE Press, 2006: 93-102.

[17] Lu Wenbin, Yang Jiahai, Liu Hongbo. Peer-to-peer node recognition algorithm based on node connection mode [J]. Journal of Tsinghua University: Natural Science, 2009, 49(7): 1045-1049.

[18] Nguyen T, Armitage G. A survey of techniques for internet traffic classification using machine learning [J]. IEEE Communications Surveys & Tutorials, 2008, 10(4): 56-76.

[19] Li Ling, Liu Huawen, Xu Xiaodan, et al. A multi-label feature selection algorithm based on information gain [J]. Computer Science, 2015, 42(7): 52-56.

[20] Xie G, Iliofotou M, Keralapura R, et al. Subflow: towards practical flow-level traffic classification [C]// Proc of IEEE INFOCOM. [S. l.]: IEEE Press, 2012: 2541-2545.

[21] Birant D, Kut A. ST-DBSCAN: An algorithm for clustering spatial-temporal data [J]. Data & Knowledge Engineering, 2007, 60(1): 208-221.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv —Machine translation. Verify with original.