Peer-to-Peer Application Recognition Based on Signaling Activity

Chen-Chi Wu[†], Kuan-Ta Chen[‡], Yu-Chun Chang[†], and Chin-Laung Lei[†]
[†]Department of Electrical Engineering, National Taiwan University
[‡]Institute of Information Science, Academia Sinica
{bipa,congo}@fractal.ee.ntu.edu.tw, ktchen@iis.sinica.edu.tw, lei@cc.ee.ntu.edu.tw

***Abstract*—Because of the enormous growth in the number of peer-to-peer (P2P) applications in recent years, P2P traffic now constitutes a substantial proportion of Internet traffic. The ability to accurately identify different P2P applications from the network traffic is essential for managing a number of network traffic issues, such as service differentiation and capacity planning. However, modern P2P applications often use proprietary protocols, dynamic port numbers, and packet encryptions, which make traditional identification approaches like port-based or signature-based identification less effective.**

In this paper, we propose an approach for accurately recognizing P2P applications running on monitored hosts based on *signaling behavior*, which is regulated by the underlying P2P protocol; therefore, each application possesses a distinguishing characteristic. We consider that the signaling behavior of each P2P application can serve as a unique signature for application identification. Our approach is particularly useful for three reasons: 1) it does not need to access the packet payload; 2) it recognizes applications based purely on their signaling behavior; and 3) it can identify particular P2P applications. The performance evaluation shows that 92% of a real-life traffic trace can be correctly recognized within a 5-minute monitoring period.

***Index Terms*—Application identification, BitTorrent, Skype, Support vector machine, Traffic classification**

I. Introduction

Peer-to-peer (P2P) traffic now constitutes a substantial proportion of Internet traffic. The ability to identify different P2P applications from the network traffic is important for network management functions, such as service differentiation, capacity planning, and QoS provisioning. For example, network administrators limit or block P2P traffic that occupies a high amount of bandwidth to ensure that other applications have sufficient bandwidth. Another issue is that most content shared in P2P networks infringes copyright laws.

Managing P2P traffic is a major challenge for network administrators because P2P applications tend to use *dynamic port numbers* and *proprietary protocols*. In this paper, we propose a model that can *recognize particular P2P applications running on the monitored host without examining packet payloads*. The key to our approach is recognizing *the signaling behavior* of a P2P application. Although a P2P application can easily change its port number, payload, and even message format, the signaling patterns between peers are more fundamental and are therefore unlikely to change. For example, a BitTorrent client needs to regularly exchange the file bitmaps containing the block status of its files with neighboring peers. This signaling behavior is essential for the maintenance of the BitTorrent network, so changing it would lead to state inconsistency and software incompatibility problems. Hence, it would be difficult for a P2P application to change its signaling behavior without affecting its normal operations. Moreover, the signaling behavior is regulated by the underlying P2P protocol, so each application possesses distinctive features. We consider that the signaling behavior of each P2P application can serve as a unique signature for application identification.

The contribution of our approach is threefold:

- It recognizes P2P applications based on their unique signaling behavior, rather than by examining the packet payload. Since this behavior is relatively difficult to change compared to the port numbers or packet format, it is unlikely that an application will be able to evade recognition.
- It only needs to examine traffic associated with the monitored host; in other words, a global view of the network is not necessary.
- It can recognize particular applications running on the monitored host, so it does not treat all P2P traffic in the same way.

The remainder of the paper is organized as follows. In Section II, we review related works. We describe the data collection methodology and summarize our traces in Section III. In Section IV, we discuss the fundamental concepts behind our approach. We present a detailed description of our scheme in Section V and evaluate the scheme's performance in Section VI. Then, in Section VII, we summarize our conclusions.

II. Related Work

In recent years, a number of works have addressed the issue of P2P traffic identification. Early approaches relied on the port numbers used by applications [10], but the estimates are now regarded as misleading because P2P applications may use dynamic ports or the default ports of other applications (e.g., port 80 or 443).

The application-layer approach identifies a protocol-specific signature by examining the packet payload [2, 9]. This ap-

This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC97-2219-E-001-001 and NSC97-2219-E-011-006. It was also supported in part by the National Science Council of Taiwan under the grant NSC96-2628-E-001-027-MY3.

TABLE I
TRACE SUMMARY

Data set	Time (hr)	Hosts	Packets	Bytes (MB)
Set 1				
- BitTorrent	410	$110,711$	$104,722,150$	594
- eMule	337	$42,377$	$36,716,588$	363
- Skype	325	$61,777$	$34,076,328$	354
- WoW	26	218	$2,528,359$	680
- TELNET	15	362	$21,118,522$	$7,331$
- HTTP	2	$4,448$	$28,264,360$	$31,097$
Set 2	0.65	$61,646$	$91,286,727$	$60,163$

proach can recognize particular P2P applications and achieve high detection accuracy because false positives do not occur if the signature is sufficiently unique; however, it cannot identify applications with unknown signatures and it cannot be used on encrypted traffic. Furthermore, examining user payloads raises privacy and legal concerns. The high computation overhead for checking signatures is another drawback of this approach.

To overcome the limitations of port-based and application-layer approaches, numerous works employ transport layer information to identify traffic. In [5], Karagiannis et al. use connection patterns to identify P2P traffic flows. Their method first searches for source-destination IP pairs that have established both TCP and UDP connections, and then treats the relationship between the number of distinct IP addresses and ports as a signature for identifying P2P traffic. Different transport-layer characteristics of P2P applications are used for identification in [4, 7, 8]. These works focus on identifying P2P traffic at the network level and do not try to recognize particular P2P applications associated with the monitored traffic.

The concept of our approach is similar to that in [6], whereby the host's behavior is analyzed at the transport layer across three levels, namely the social level, the functional level, and the application level. This method identifies the type of an application, e.g., P2P, web, or gaming; however, our goal is to identify distinct P2P applications, such as Gnutella, eMule, or BitTorrent.

III. DATA DESCRIPTION

Here, we introduce the two sets of traffic traces used in this work. The first set (*Set 1*) was captured on end hosts or gateway routers, and the second (*Set 2*) was captured from full packet payloads on a link that connects a campus network to the Internet. We use *Set 1* to study the behavior of P2P applications and evaluate our scheme. Then, *Set 2*, which reflects the traffic composition of the campus network, is used to validate the identification result of our scheme by comparing it with that of a payload-based approach.

Set 1 consists of six types of network applications: BitTorrent, eMule, Skype, online games, TELNET, and HTTP. We captured the traces of the first three applications on three hosts that execute BitTorrent, eMule, and Skype respectively. Specifically, we used BitTorrent version 6.0.3, eMule version 0.48a, and Skype version 3.6.1, which were the latest versions published on the official web sites just before the trace collection process started on March 14, 2008. We varied the configuration settings during the collection period because some settings, such as the number of connections, may affect the signaling behavior of applications. For the BitTorrent and eMule clients, the maximum number of simultaneous connections was randomly set to a value between $2,000$ and $3,500$; and the maximum number of simultaneously connected hosts per download file was randomly set to a value between 150 and 250 for each trace.

We executed WinDump on each host to capture its traffic. Since our objective is to recognize P2P applications based purely on their signaling behavior, we only require the signaling packets sent from and received by the monitored hosts. As distinguishing between signaling packets and data transfer packets requires application-specific knowledge and payload dissection, we simply assumed that packets with a payload size smaller than 100 bytes were signaling packets and only collected such packets.

TELNET traffic was captured on a gateway router for all TCP flows with port numbers 22 (SSH) and 23 (telnet); and all intra-campus traffic was removed. For online games, we chose World of Warcraft and collected the traffic on a gateway router for all TCP flows with port number 3274; either the source or the destination address is within the network 203.66 (where the World of Warcraft server is located in Taiwan).

Set 2 was collected from a campus network on February 26, 2007. To analyze this data set, we use an application-layer classifier that identifies characteristic patterns in the packet payload[1]. According to the classification results, 32.7% of the packets and 12.9% of the bytes were from P2P applications (BitTorrent, eMule, and Skype). The remaining packets and bytes were from other categories (e.g., web, chatting, gaming). Table I summarizes the traces in *Set 1* and *Set 2*.

IV. FUNDAMENTALS OF THE PROPOSED SCHEME

In this section, we present the rationale for our scheme, which is based on empirical observations. P2P applications generate two types of traffic: data transfer traffic and signaling traffic. Data transfer traffic refers to the file-sharing or file-redistribution traffic transmitted between peers in the network. On the other hand, signaling traffic is used for updating file information, peer discovery, probing the path quality, and the exchange of other control information. To verify our conjecture that each application has a unique characteristic, in the following, we first compare the signaling traffic statistics of BitTorrent, eMule, and Skype. We then illustrate and contrast the differences in the signaling patterns of the three applications.

A. Signaling Traffic Statistics

In Fig. 1(a)-(c), we plot the number of hosts that exchange signaling packets with the monitored hosts in a 2-hour period. A host is considered old if it has been observed sending/receiving packets within 5 minutes; otherwise, it is regarded as a new host. In Figure 1(a), for BitTorrent clients, the number of hosts contacted by the monitored host increases steadily over time, but stops increasing after 40 minutes. One possible explanation is that the maximum number of connections bounds the number of hosts. As shown in Fig. 1(b),

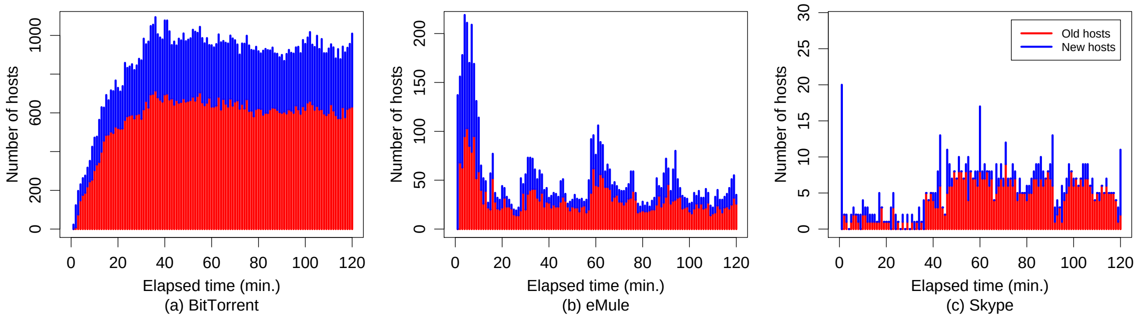

Fig. 1. The signaling traffic statistics: the number of old hosts and new hosts contacted by the monitored hosts.

the monitored host contacts a large number of hosts initially, but the number of hosts decreases dramatically after about 10 minutes and continues to fluctuate over time. Fig. 1(c) shows that the Skype host contacts a dozen hosts during the login process and contacts a near constant number of hosts after 60 minutes.

Next, we analyze the ratio of new hosts to all hosts contacted by the monitored host. Compared to the eMule and Skype hosts, we observe that the BitTorrent host has a high ratio of new hosts most of the time. On the other hand, the ratio of new hosts is only high in the early stage for eMule; and the Skype host only has a high ratio of new hosts during the login process. The above preliminary analysis evidences that BitTorrent clients may continuously seek new hosts, whereas eMule and Skype clients tend to contact old hosts.

Based on the above analysis, we show that different P2P applications exhibit very different signaling behavior in terms of the number of hosts and the ratio of new and old hosts.

B. Signaling Patterns

In Figure 2, we plot the signaling patterns of BitTorrent, eMule, and Skype in a 1-hour period. We assign numeric identifiers to hosts that have been contacted by the monitored host based on the order in which they are observed, and use the sign of the identifiers to denote the direction of the signaling traffic. A positive identifier indicates a packet sent from the monitored host to the peer host, while a negative identifier indicates a packet sent from the peer host to the monitored host. Each dot in Figure 2 represents a signaling packet sent from or received by the monitored host. An intensive exchange of signaling packets is depicted by a high distribution of dots or even an area of solid color in this figure. In the following, we discuss the unique characteristics of each application.

BitTorrent: The high dot density in Fig. 2(a) implies an intensive exchange of signaling packets between the monitored BitTorrent client and its peer hosts. Moreover, the near linear growth of the number of peer hosts shows how a BitTorrent client progressively discovered new hosts during the monitoring period.

eMule: As shown in Fig. 2(b), the number of signaling packets sent from and received by the eMule client is much fewer than those of the BitTorrent client. We also observe that the number of peer hosts increases rapidly in the first 10 minutes, but it increases slowly thereafter.

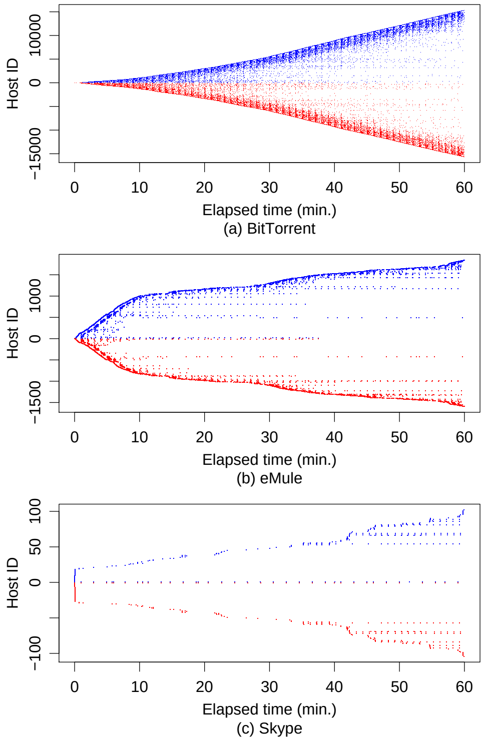

Fig. 2. The signaling activity patterns of BitTorrent, eMule, and Skype.

Skype: The sparse distribution of dots in Fig. 2(c) suggests that most of the signaling packets belonged to probe traffic, which is a pair of packets comprised of a single probe packet and the corresponding reply packet [3], i.e., a packet with a host ID X is coupled with another packet with the host ID -X.

The above graphical comparisons show that each P2P application possesses a number of unique signaling characteristics. In the next section, we utilize the distinctiveness of each application's signaling patterns to develop a P2P application recognition scheme.

V. The Proposed Scheme

In this section, we propose a P2P application identification scheme, which is based on the signaling traffic associated with the monitored host . First, we explain how we characterize the signaling behavior and how we derive features from signaling packet streams. We then exploit the features to design a classifier for recognizing individual P2P applications.

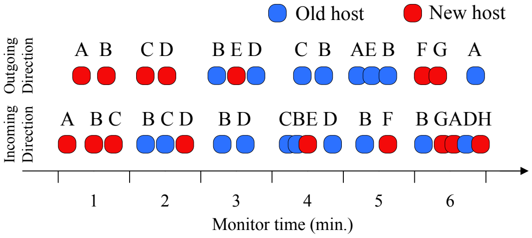

Fig. 3. A simple scenario of signaling behavior at the host level.

TABLE II
FEATURES USED TO CHARACTERIZE THE SIGNALING BEHAVIOR OF PEER-TO-PEER APPLICATIONS.

Host level
Ratio of new / old hosts (mean, sd[†])
Growth rate of new / old hosts (mean, sd)
Correlation coefficient between the number of new and old hosts
Message level
Ratio of new / old packets (mean, sd)
Growth rate of new / old packets (mean, sd)
Correlation coefficient between the number of new and old packets
Alternate rate of new and old packets (mean, sd)

[†] Standard deviation

A. Signaling Behavior Characterization

Since an application's signaling behavior is regulated by its underlying P2P protocol, each application possesses a distinguishing characteristic. Based on this concept, we keep track of all the signaling packets sent from and received by a monitored host, and characterize the host's signaling behavior on two levels: the host level and the message level.

Host level: A host regularly exchanges information with other hosts that are known to it, and also with new contacts. Based on the number of new or old hosts the monitored host communicates with, we can characterize the signaling behavior at the host level.

Message level: For the monitored host, we denote a signaling packet exchanged with a new host as a new packet; otherwise, it is regarded as an old packet. Based on the number of new or old signaling packets, we define a number of features to represent the signaling behavior at the message level.

Our approach monitors a host for a certain period and then uses the derived features to recognize the P2P applications running on the host. For each monitored host, we count the number of hosts contacted and the number of packets sent and received every minute. *We assume that a host is an old host if it has been observed sending/receiving packets within the previous* 5 *minutes; otherwise, it is considered a new host.* In the following, we describe how we derive features at the host level.

Ratio of new/old hosts: At the end of the monitoring period, we obtain two sets of values for the ratio of new/old hosts per minute , and then calculate the mean and standard deviation of the ratio of new/old hosts. For example, in Figure 3, five hosts communicate with the monitored host about the incoming traffic in the 6th minute. Since hosts B and D were observed in the 4th and 5th minutes respectively, they are labeled old hosts; on the other hand, hosts A, H, and I are considered new hosts. Therefore, the ratio of new hosts and old hosts in the 6th minute is $3/5$ and $2/5$ respectively.

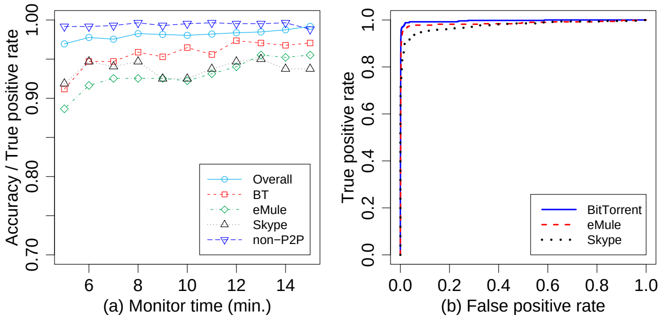

Fig. 4. The influence of the length of the monitoring period on the accuracy and true positive rate, and the ROC curves for the traces of P2P applications in Set 1.

Growth rate of new/old hosts: The growth rate of new/old hosts refers to the change in the number of new/old hosts over a certain period. In this paper, we compute the growth rate every two minutes. For example, in the incoming direction, the number of old hosts in the 4th and 5th minutes is 3 and 1 respectively. Thus, the growth rate of old hosts in the 4th and 5th minutes is $(1-3)/3 = -0.67$. At the end of the monitoring period, the mean and standard deviations of the growth rate can be inferred from the values we compute every two minutes.

Correlation coefficient between the number of new and old hosts: The correlation coefficient is derived to characterize the correlation between the number of new and old hosts. In Fig. 3, the series of the number of new and old hosts in the outgoing direction is $(2, 2, 1, 1, 1, 2)$ and $(0, 1, 2, 2, 2, 1)$ respectively. From these two series, we derive a correlation coefficient, which indicates the degree of linear dependence between the number of new and old hosts.

We derive the message-level features in the same way as the host-level features. The features we use are summarized in Table II. Each feature comprises a pair of values computed from traffic in both directions. We also compute the correlation between the values of each pair.

B. Classifier Design

We use a support vector machine (SVM), a supervised machine learning method, to identify P2P applications. Our scheme comprises two phases: a training phase and a recognition phase. In the training phase, we derive features from each training stream of signaling packets, and then apply an SVM to train the classifier. In the recognition phase, given a signaling packet stream, we extract its features and use the trained classifier to determine which P2P application is associated with the stream.

VI. PERFORMANCE EVALUATION

We now evaluate the performance of our scheme with 10-fold cross validation based on the traces described in Section III.

First, using the traces in *Set 1*, we discuss the effect of the length of the monitoring period on the true positive rate and false positive rate for each application. As shown in Fig. 4(a),

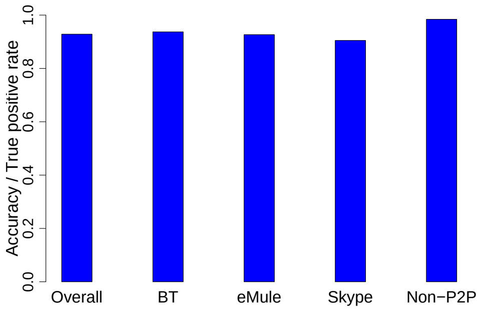

Fig. 5. The overall accuracy and true positive rate of each category for the traces in Set 2.

our classifier achieves over 96% accuracy in a 5-minute monitoring period and 98% accuracy in an 8-minute period. It achieves a true positive rate of 90% for each application with a monitoring period longer than 6 minutes. We also observe that the true positive rate of the non-P2P category is nearly 99%, which means our scheme seldom misidentifies a non-P2P sample as a P2P sample. Furthermore, the false positive rate of each category is lower than 5%. For an overall evaluation of the traces in *Set 1*, we plot the ROC curves for each P2P application, as shown in Fig. 4(b). The ROC curve of each application depicts its performance by treating the application as the positive class and all other applications as the negative class. We observe that each ROC curve in Fig. 4(b) passes through the upper left corner of the plot (i.e., the true positive rate is close to 1 with a small false positive rate). Therefore, these curves evidence that our proposed scheme can recognize each application with a high true positive rate and an extremely low false positive rate.

Next, we use the classification result of an application-layer classifier as a reference point to evaluate the performance of our scheme based on the traces in *Set 2*. Based on the application-layer classification result, unknown flows are excluded from the data set. We also perform 10-fold cross validation and set the monitoring period to 5 minutes. As shown in Fig 5, our classifier achieves 92% overall accuracy and the true positive rate of each category is over 90%. The above identification results for both data sets demonstrate that our scheme can recognize P2P applications running on monitored hosts with a high degree of accuracy.

VII. Conclusion and Future Work

In this paper, we propose a scheme for recognizing P2P applications based on the signaling behavior of the applications. Our approach is particularly useful in two respects. First, it does not need to access the packet payload or rely on port numbers. It only needs to analyze the signaling traffic associated with the monitored host. Second, it can recognize particular P2P applications running on the monitored host. By analyzing the signaling behavior at the host and message levels, we show that 92% of a real-life trace can be correctly recognized in a 5-minute monitoring period.

Although the evaluation results are promising, we will improve the following two aspects of our scheme in a future work.

Launching multiple P2P applications on a host. When a host launches multiple P2P applications, all signaling traffic is combined. As a result, the proposed scheme cannot recognize particular applications based purely on observations of all the signaling traffic of the host. To solve this problem, we are working on demultiplexing traffic that consists of signaling packets generated by more than one P2P application. Based on the port numbers used by each traffic flow, we gather flows that use related port numbers into a group. We assume that each group of traffic flows only contains signaling traffic generated by a single application. With this heuristic, our scheme can recognize individual applications by analyzing each group of traffic flows.

Short flows. Since the monitoring period of the proposed scheme is at least 5 minutes, we are unable to correctly detect P2P applications with signaling traffic that is shorter than 5 minutes. For example, a user may make a phone call that lasts less than 5 minutes and sign out immediately after finishing the call. In this case, we cannot collect enough signaling traffic to derive the signaling features. In our future work, we will devise other powerful features to characterize the signaling behavior of short P2P sessions.

References

[1] "L7-filter Supported Protocols," http://l7-filter.sourceforge.net/protocols/.

[2] H. Bleul, E. P. Rathgeb, and S. Zilling, "Evaluation of an efficient measurement concept for p2p multiprotocol traffic analysis," in *EUROMICRO '06: Proceedings of the 32nd EUROMICRO Conference on Software Engineering and Advanced Applications*. Cavtat, Croatia: IEEE Computer Society, 2006, pp. 414–423.

[3] D. Bonfiglio, M. Mellia, M. Meo, N. Ritacca, and D. Rossi, "Tracking down skype traffic," in *Proceedings of IEEE INFOCOM'08*, Phoenix, AZ, USA, 2008, pp. 15–17.

[4] F. Constantinou and P. Mavrommatis, "Identifying known and unknown peer-to-peer traffic," in *NCA '06: Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications*, Cambridge, MA, USA, 2006, pp. 93–102.

[5] T. Karagiannis, A. Broido, M. Faloutsos, and K. claffy, "Transport layer identification of p2p traffic," in *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, Taormina, Sicily, Italy, 2004, pp. 121–134.

[6] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," in *Proceedings of ACM SIGCOMM'05*, Philadelphia, Pennsylvania, USA, 2005, pp. 229–240.

[7] X. Lu, H. Duan, and X. Li, "Identification of p2p traffic based on the content redistribution characteristic," in *ISCIT'07: Proceedings of the International Symposium on Communications and Information Technologies*, Sydney, Australia, 2007, pp. 596–601.

[8] M. Perenyi, A. G. Trang Dinh Dang, and S. Molnar, "Identification and analysis of peer-to-peer traffic," *Journal of Communication*, vol. 1, no. 7, pp. 36–46, 2006.

[9] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," in *WWW '04: Proceedings of the 13th international conference on World Wide Web*, New York, NY, USA, 2004, pp. 512–521.

[10] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 12, no. 2, pp. 219–232, 2004.