Multi-Level P2P Traffic Classification Using Heuristic and Statistical-Based Techniques: A Hybrid Approach

Max Bhatia 1, Vikrant Sharma 1, Parminder Singh 1,*, and Mehedi Masud 2,*

1 Department of Computer Science Engineering, Lovely Professional University, Punjab 144001, India; max.16870@lpu.co.in (M.B.); vikrant.23738@lpu.co.in (V.S.)
2 Department of Computer Science, College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
* Correspondence: parminder.16479@lpu.co.in (P.S.); mmasud@tu.edu.sa (M.M.)

Received: 31 October 2020; Accepted: 17 December 2020; Published: 20 December 2020

Abstract: Peer-to-peer (P2P) applications have been popular among users for more than a decade. They consume a lot of network bandwidth, due to the fact that network administrators face several issues such as congestion, security, managing resources, etc. Hence, its accurate classification will allow them to maintain a Quality of Service for various applications. Conventional classification techniques, i.e., port-based and payload-based techniques alone, have proved ineffective in accurately classifying P2P traffic as they possess significant limitations. As new P2P applications keep emerging and existing applications change their communication patterns, a single classification approach may not be sufficient to classify P2P traffic with high accuracy. Therefore, a multi-level P2P traffic classification technique is proposed in this paper, which utilizes the benefits of both heuristic and statistical-based techniques. By analyzing the behavior of various P2P applications, some heuristic rules have been proposed to classify P2P traffic. The traffic which remains unclassified as P2P undergoes further analysis, where statistical-features of traffic are used with the C4.5 decision tree for P2P classification. The proposed technique classifies P2P traffic with high accuracy (i.e., 98.30%), works with both TCP and UDP traffic, and is not affected even if the traffic is encrypted.

Keywords: heuristic-based classification; multi-level P2P traffic classification; P2P-port based classification; statistical-based classification

1. Introduction

The P2P networking technology is used to share and distribute media, documents, software, etc., among peers. A decade ago, peers on the Internet used the client-server architecture, where the clients request data from the server and the server responds with the requested data. Due to this reason, the majority of the Internet traffic used to be asymmetric in nature. However, with the evolution of P2P traffic, network traffic started becoming symmetric. In such a case, a peer starts acting simultaneously as a client and server, thereby downloading and uploading the data at the same time. Due to this factor, as well as a rise in the number of P2P users, it has become one of the major contributors of internet traffic. It has ended the dominance of other numerous application protocols (for example, FTP, SMTP, HTTP, etc.), which used to rule the Internet more than a decade ago [1]. There has been a significant trend of P2P file-sharing, in recent years, through P2P applications where audios, videos, games, and software are being shared or distributed, significantly large in size [2].

The main issue with P2P traffic is that it consumes a large amount of network bandwidth [1,3–5]. Conventional network devices cannot handle the traffic of P2P applications, due to the fact that network administrators and ISPs face various challenges such as providing excellent broadband experience to customers, purchasing of backbone links, and upstreaming bandwidth, which are costly.
The main issue with P2P traffic is that it consumes a large amount of network bandwidth. Due to this reason, other kinds of application protocols do not get a fair amount of network bandwidth, resulting in a poor Quality of Service for such applications. Therefore, it is required to monitor and classify P2P traffic, which will help ISPs and network administrators perform various tasks, for example, implementing network specific policies for providing Quality of Service to each network application, implementing billing mechanisms based on the type of traffic used by customers, implementing network security measures, addressing network congestion issues, etc. Furthermore, enterprises can either limit or ban P2P traffic from evading network congestion and maintaining Quality of Service for various applications in their network, as shown in Figure 1.

![Figure 1. Controlling the quality of service.](image)

Nowadays, classifying P2P traffic accurately is a difficult task as various P2P applications, either masquerade or encrypt their traffic to avoid detection [6,7]. There are some techniques for classifying the network traffic, such as port-based, payload-based, and Classification in the Dark (which includes statistical-based, pattern, or heuristic-based techniques) [7–9]. Since many P2P applications are masquerading their traffic either by disguising port numbers or encrypting payloads, port-based and payload-based techniques are inefficient in accurately classifying P2P traffic. Classification in the Dark techniques rely on the traffic’s statistical features or behavioral patterns to perform the classification and hence, do not rely on port numbers or payload contents of the traffic. They are effective in classifying P2P traffic these days. They can also classify encrypted traffic and unknown applications from target classes but cannot perform the traffic classification with high accuracy as the payload-based technique [6]. Therefore, to achieve a high classification accuracy of P2P traffic, a single method alone may not be sufficient. We propose a hybrid technique in this paper, which is the combination of heuristic-based and statistical-based techniques.

The main aim of this paper is to propose a hybrid technique for P2P traffic classification, which accomplishes the following tasks:

- ability to classify P2P traffic with high accuracy.
- ability to work with both TCP and UDP protocols (since various P2P applications use either TCP or UDP or both protocols for communication).
• involves less computation in classifying P2P traffic (by not relying on the DPI approach for classification) in comparison to various existing hybrid techniques.
• ability to classify P2P traffic even if it is encrypted.

The experiments performed using the proposed hybrid technique achieved a high classification accuracy of 98.30%, which is higher than other hybrid/non-hybrid techniques, and also combines the benefits of heuristic-based (less computation as compared to DPI) as well as statistical-based (scalability) techniques. Further, unlike various existing hybrid techniques, the proposed technique does not rely on the signature-based technique. Rather, it utilizes a set of heuristic rules which comparatively involves less computation in classifying P2P traffic. In addition to that, the heuristics proposed in this paper perform equally with both TCP and UDP traffic flows and are not affected even if a traffic flow is encrypted.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 analyzes various P2P traffic classification techniques. Section 4 discusses the multi-level P2P traffic classification technique. Section 5 discusses the evaluation criteria and experimental results. Finally, Section 6 concludes the research work.

2. Related Work

P2P applications have become very popular since the past decade, and the traffic generated by such applications continues to grow as new applications keep emerging and many peers join the network to use them. P2P traffic is one the largest contributors to internet traffic [10], which has a major impact on it due to its large volume and long connection time, leading to network congestion. Its traffic flows in large amounts in both directions, i.e., P2P applications act as a client and server concurrently by downloading the data from other peers and serving the request of multiple other peers by uploading the data requested. P2P applications are generally utilized for sharing large files among various peers. Once initiated, these applications require little or no human intervention and are usually left unattended for a long time, which results in a large network activity throughout the day [11]. Therefore, such kind of traffic can be observed naturally over 24 h.

Conventional traffic classification techniques such as port-based and payload-based are ineffective in classifying P2P traffic due to the various limitations associated with them. Hence, modern classification techniques such as statistical-based or heuristic-based are employed for this purpose. Reddy and Hota [12] used the heuristic-based technique by analyzing connection patterns of the host to identify P2P traffic and found the average detection rate of 99%. They achieved this detection rate by classifying the TCP flows as non-P2P, which communicates over default port number 80. However, if a P2P application masquerades using a TCP port number (e.g., 80 used by HTTP) [5] or a new P2P application protocol emerges with different communication patterns, then it may not satisfy any of the proposed heuristics. Hence, it would lead to many miss-classifications, due to which a high detection rate may not be achieved. Bozdogan et al. [13] assessed four supervised and one un-supervised ML algorithms, namely SVM, C4.5 decision tree, Ripper, Naïve Bayesian, and K-means, respectively, for the identification of P2P network applications. They found that Ripper and C4.5 algorithms have a similar performance with the detection rate ranging between 58.9–99.1% and 15.6–98.1%, respectively. However, the evaluation was performed using only three P2P applications namely BitComment, BitTorrent, and uTorrent. Tseng et al. [14] proposed a methodology to classify P2P traffic on the basis of aggregation clustering. A similar traffic flow was aggregated by determining the correlation between clusters through their distance ratio. This approach classifies both known and unknown traffic flows with an overall accuracy of 90.50%. Chuan et al. [15] utilized the Bat algorithm to search the most relevant parameters, which can be used with SVM for classifying P2P traffic and was able to achieve the classification accuracy ranging between 86.77–91.34%. Abdalla et al. [16] proposed a multi-stage method for feature selection in order to create a subset of optimal statistical traffic features that can be utilized for online classification of P2P traffic. The authors used J48 and Naïve Bayes as ML algorithms, which achieved classification accuracy and recall rates ranging between
96.29–99.78% and 86.9–99.8%, respectively using a set of six proposed features. However, these six proposed features alone may not be effective in classifying existing P2P applications, which may have evolved (since the creation of public datasets which are used here) or newer P2P application protocols as they emerge. Jamil et al. [17] proposed an approach to develop a model which combines SNORT rules (which is based on the packet payload) and the ML algorithm for classifying P2P traffic. The technique used fuzzy-rough and Chi-square as feature selection algorithms and evaluated the performance of 3 ML algorithms namely SVM, C4.5 decision tree, and ANN and achieved a 99.7% classification accuracy using the combination of ANN and C4.5. However, the technique relies on the payload-based approach (SNORT), which has various limitations. Nazari et al. [18] proposed an approach called DSCA, which is based on the DPI technique for the identification of various P2P and non-P2P applications over an encrypted network. The proposed technique used four modules, namely feature-extractor (for maintaining the flows), inline-DPI (for labelling traffic flows and detecting new applications), stream-processor (for handling flows between the feature-extractor and stream-classifier), and stream-classifier (for building the classification function). The experimental results achieved a maximum classification accuracy of 96.75%. However, this technique also relies on the payload based approach, which has various limitations. Ye and Cho [19–21] proposed a hybrid technique to classify P2P traffic in two steps. The first step performs classification at the packet-level by combining signature-based and heuristic-based techniques. The second step performs classification at the flow-level by combining statistical-based and heuristic-based techniques to classify the remaining unknown traffic. The authors achieved an overall flow-accuracy and byte-accuracy ranging between 97.70–98.19% and 97.06–99.82%, respectively. However, their technique does not classify the UDP traffic and also relies on the payload-based approach (which has various limitations). Khan et al. [22] proposed a hybrid approach for classifying the traffic into normal P2P and P2P-botnet. In the first stage, the non-P2P traffic is separated by using the mechanism of well-known port numbers, DNS query filtering, and flow-counting rules. The remaining traffic is considered as P2P traffic and is fed into the second stage where the wrapper method is utilized for selecting traffic features and the decision tree algorithm is employed for classifying the traffic either as normal P2P or P2P-botnet. The experimental results achieved the classification accuracy of 94.4%. However, this technique considers the network traffic to be non-P2P (in the first stage), which uses well-known port numbers (e.g., 20, 21, 80, 443, etc.) for communication. This could lead to many false negative cases, since many P2P applications can masquerade using these well-known port numbers and hence, such traffic can go undetected.

A multi-level P2P traffic classification technique is proposed in this paper, which is a hybrid approach. It utilizes the combination of heuristic-based and statistical-based techniques for classifying P2P traffic. In addition, it does not rely on the payload-based technique for classification (which has various limitations), but rather utilizes a set of heuristic rules proposed in this paper, which comparatively involves less computation, performs equally with both TCP and UDP traffic flows, and is not affected even if the traffic flow is encrypted.

3. Analysis of Existing P2P Traffic Classification Techniques

Earlier, the task of classifying Internet traffic was easy and simple as it required the information of port numbers only to perform the traffic classification. However, as P2P applications evolved, they started masquerading the traffic using well-known port numbers (for example: HTTP, HTTPS, etc.) or random port numbers to avert detection. Therefore, the port-based technique started becoming inefficient. As a result, another technique was utilized, which is based on the packet payload of the traffic for its classification. However, this technique also has its fair share of limitations, due to the fact that newer classification techniques are adopted currently, which are either statistically based, pattern or heuristic-based, or even a hybrid technique to overcome the drawbacks of traditional classification techniques. A brief description of these techniques along with the related work is mentioned below.
3.1. Port-Based Traffic Classification

This technique classifies the network traffic based on the TCP/UDP port number present in the transport layer of a packet header. Internet assigned numbers authority (IANA) [23] defines well-known port numbers, which are associated with each application protocol. For example, FTP traffic is transferred using port number 20 and 21, HTTP traffic is transferred using port number 80, SMTP traffic is transferred using port number 25, etc. In this technique, the TCP/UDP port number is extracted from the packet header. In the case of TCP connection, a classifier analyzes the SYN packets (packets which are used for the three-way handshake for establishing the connection) target port number from the registered list of port numbers defined by IANA [23], for classifying the network traffic to a particular type. Similarly, the UDP traffic is classified by using port numbers used by the hosts during communication, but unlike TCP, it does not involve any connection establishment. Gomes et al. [6] mentioned several well-known port numbers used by various P2P application protocols, some of which are shown in Table 1. The prime advantage of this classification technique lies in its simplicity to implement as it does not involve any calculations to classify network traffic. For newer applications, only the database is required to be updated with new port numbers. However, with the proliferation of the Internet, various P2P applications have emerged and evolved. They either use random port numbers for communication (which may not be registered with IANA) [24], or use the masquerading technique to disguise their traffic so that they appear to originate from a well-known protocol (e.g., HTTP, HTTPS, etc., which is not blocked or filtered) and hence, such kind of traffic goes undetected. Therefore, the port-based technique is inefficient in classifying all P2P traffic correctly and has become obsolete now [25–27].

Table 1. List of well-known ports used by various peer-to-peer (P2P) protocols.

<table>
<thead>
<tr>
<th>Protocols</th>
<th>TCP/UDP Port Numbers</th>
</tr>
</thead>
<tbody>
<tr>
<td>BitTorrent</td>
<td>6881–6999</td>
</tr>
<tr>
<td>Direct Connect</td>
<td>411, 412, 1025–32,000</td>
</tr>
<tr>
<td>eDonkey</td>
<td>2323, 3306, 4242, 4500, 4501, 4661–4674, 4677, 4678, 4711, 4712, 7778</td>
</tr>
<tr>
<td>FastTrack</td>
<td>1214, 1215, 1331, 1337, 1683, 4329</td>
</tr>
<tr>
<td>Yahoo (messages/video/voice)</td>
<td>5000–5010, 5050, 5100</td>
</tr>
<tr>
<td>Napster</td>
<td>5555, 6257, 6666, 6677, 6688, 6699–6701</td>
</tr>
<tr>
<td>MSN (voice/file-transfer)</td>
<td>1863, 6891–6901</td>
</tr>
<tr>
<td>MP2P</td>
<td>10,240–20,480, 22,321, 41,170</td>
</tr>
<tr>
<td>Kazaa</td>
<td>1214</td>
</tr>
<tr>
<td>Gnutella</td>
<td>6346–6347</td>
</tr>
<tr>
<td>ARES Galaxy</td>
<td>32285</td>
</tr>
<tr>
<td>AIM (messages/video)</td>
<td>1024–5000, 5190</td>
</tr>
</tbody>
</table>

Madhukar and Williamson [28] showed that the port-based technique could not classify internet traffic correctly. Karagiannis et al. [29] found that 30% to 70% of the traffic used random port numbers, which were generated by P2P applications and various P2P applications used the well-known port 80 (i.e., HTTP) for transferring their data. Moore and Papagiannaki [26] could achieve a maximum of 70% byte accuracy using the port-based classification technique. Since the port-based traffic classification is a conventional technique, its related work is referred to in [6].

3.2. Payload-Based Traffic Classification

This technique (also known as deep packet inspection or DPI) makes use of packet payload to classify network traffic. It utilizes the database containing signatures of application protocols which
have been stored previously. In addition, it inspects the packet payload of the traffic bit-wise to locate a bit-stream containing pre-defined byte sequence (called signatures) of the application protocol. In this way, the traffic is classified accurately when the packet-signatures extracted from network application maps with one of the packet-signatures are already stored in the database. For example, ‘\GET’ string is found in HTTP traffic, ‘xe3\x8’ string is found in eDonkey P2P traffic, etc. The prime advantage of this technique is classifying the network traffic with very high accuracy. However, this technique also has various limitations [6,30,31], which are mentioned below:

- it involves a great amount of processing load and complexity.
- it is not feasible in high-speed networks as it needs a large amount of computational resources for inspecting the traffic.
- it is almost impossible to classify the network traffic if it contains a proprietary protocol or if the traffic is encrypted.
- it may violate privacy policies of some organizations which do not allow a direct inspection of packets.

Since the payload-based traffic classification is a conventional technique, its related work is referred to in [6].

3.3. Classification in the Dark

This approach overcomes the drawbacks of port-based and payload-based techniques (as mentioned in previous sections). It classifies network traffic by either using statistical properties of packets associated with a traffic flow [25] (known as statistical-based technique), or by analyzing communication patterns of a traffic flow (known as heuristic-based technique). However, this technique does not give as high accurate results as the payload-based technique.

The basic idea behind the heuristic-based technique is to classify the network traffic using a pre-defined set of heuristic rules. These rules are constructed by analysing the communication patterns of the traffic, for example, a number of outbound connections made by the host for communication, whether the host is acting as a client and server concurrently during the communication, etc. Perenyi et al. [32] proposed a technique to identify P2P traffic using a set of six heuristics, which achieved a 99.14% recall rate. However, they cannot achieve high classification results currently, since they were made to work with the older dataset consisting of P2P applications, which are either phased-out or have already evolved since then. Yan et al. [33] proposed a novel approach based on flow statistics and host-based heuristics to identify P2P traffic and achieved a 93.9% flow accuracy and 96.3% byte accuracy. However, the results of this approach degrade if the network address translation (NAT) is in use or if the traffic uses dynamic IP addresses. Wang et al. [34] proposed a technique to utilize behavioural features of traffic flows with the C4.5 decision tree to identify P2P traffic. The experimental results achieved precision values ranging between 90.96–93.66% and recall values ranging between 86.69–95.73% in identifying PPTV, Skype, and Thunder. Zhang et al. [35] proposed the component-based technique (i.e., based on the graph theory) to analyze various P2P applications which use the UDP protocol for communication. They argue that component-level statistics can be used to detect P2P traffic reliably and accurately. However, they did not perform the experimental analysis to show its effectiveness.

The basic idea behind the statistical-based technique is that it classifies the traffic using its flow-level or packet-level properties, for example, the total bytes received/sent, packet size, packet inter-arrival time, duration of traffic flow, etc., which can be used collectively or individually to calculate statistical measures such as the average, variance, and probability density function. The assumption here is that different applications generate traffic flows that possess unique characteristics. With the increase in the number of traffic features, mapping of the traffic features with the corresponding classes manually becomes difficult. Due to this reason, ML algorithms are generally used along with statistical features of traffic. Here, a reference model is built with the help of pre-labelled training, which is then used to classify the traffic of the testing dataset. Sun and Chen [36] utilized the C4.5 decision tree for
classifying applications which use TCP flows. This technique analyzed the amount of data first sent by the hosts continuously during the communication and achieved classification accuracy ranging between 97.648–99.694%. However, it only classifies traffic associated with TCP flow and does not work on UDP flow. Gong et al. [37] proposed an incremental algorithm to improve the learning of existing SVM, which has good space and time complexity and achieved the identification accuracy of 87.89% in identifying P2P traffic. The experimental results achieved an overall classification accuracy of 92.47%. Qin et al. [39] developed a framework called CUFTI to identify P2P traffic using the payload length as well as the direction of the control packets (which appears at the start of a flow) as the flow features. However, the proposed technique used only three applications namely PPlive, BitTorrent, and Thunder for the experiment and achieved FNR and FPR rates ranging between 8.47–34.57% and 3.49–22.26%, respectively. He et al. [40] proposed a fine-grained P2P traffic classification approach by analyzing the hosts. The experimental results achieved a 97.22% true positive rate. However, the proposed technique focused only on the P2P file-sharing traffic for classification purposes. Ertam and Avci [41] used the kernel based extreme learning machine (KELM) approach combined with the genetic algorithm (GA) for feature selection in classifying the Internet traffic (which also included P2P traffic) and the experimental results achieved an average classification accuracy of 96.57%. Sun et al. [42] classified the network traffic using a model named TrAdaBoost, which is a transfer learning model and is a modified version of AdaBoost. The proposed technique classified various kinds of network traffic (such as www, mail, database, etc.) where P2P traffic was classified with a 91.8% accuracy. Lim et al. [43] utilized deep learning models namely CNN and ResNet to classify the network traffic. It used packet payloads as image data to create datasets and train deep learning models. Traffic from eight applications were used for experimental purposes, which included only two P2P applications (i.e., Skype and BitTorrent) and achieved the f1-score of 0.97.

The largest contributor of overall P2P traffic in the Internet includes file-sharing applications (e.g., BitTorrent) and VoIP applications (e.g., Skype) [10]. Therefore, there are also various studies which specifically focus on classifying such P2P applications. For example, techniques proposed in [44,45] focus on classifying the BitTorrent traffic, whereas techniques proposed in [46–51] focus on classifying the VoIP traffic.

Moreover, there are some hybrid techniques which classify P2P traffic. Li et al. [52] proposed a hybrid classification technique using the combination of C4.5 decision tree, port-based, and payload-based techniques in a two-step process and achieved an overall classification accuracy of 96.03%. Chen et al. [53] proposed a hybrid technique by combining the hardware classifier (based on the network processor) and software classifier based on FNT for classifying P2P traffic. The proposed technique achieves the accuracy of 95.67%, but it relies on a dedicated hardware for the classification of P2P traffic. Keralapura et al. [5] proposed a two-stage classifier known as SLTC (self-learning traffic classifier) to classify P2P traffic and achieved the detection rate of 95%. Nair and Sajeev [54] proposed a technique which uses the combination of pattern-based and statistical-based approaches to classify the traffic into P2P and non-P2P and achieved a maximum classification accuracy of 91.42%. The authors proposed another hybrid technique in [55], where they classified P2P traffic using the packet header and payload information in the statistical-based technique (which utilized the C4.5 ML algorithm) and achieved the detection rate of 95%

Most of the hybrid techniques discussed above classify P2P traffic by making use of the signature/payload-based technique which has various limitations (as mentioned in the previous section). Therefore, they may not be able to achieve a good classification accuracy if the traffic is encrypted or contains newer/proprietary application protocols. Apart from this, a single (non-hybrid) technique may not be sufficient for classifying P2P traffic, since depending on the approach to be utilized for classifying P2P traffic, it may not be applicable for real-time classification (due to the large computation involved) or may not be able to classify newer/proprietary application protocols [20].
Therefore, we propose a multi-level P2P traffic classification technique which is a hybrid approach. It combines heuristic-based and statistical-based techniques to achieve a high accuracy of 98.30% in classifying P2P traffic. In addition, the classification process involves less computation, since unlike other various hybrid approaches, it does not make use of the signature/payload-based technique for classifying P2P traffic.

4. Multi-Level P2P Traffic Classification Technique

Based on the previous analysis, a multi-level P2P traffic classification technique is proposed. It is divided into two steps, where the first step performs the traffic classification at a packet-level and the second step performs the traffic classification at a flow-level.

4.1. System Model for Classifying P2P Traffic

Figure 2 illustrates the overall system of the P2P traffic classification process, which is sub-divided into a two-step process, namely packet-level process and flow-level process. In the packet-level classification process, the P2P-port based technique in combination with the packet-heuristics based technique performs a traffic classification. The traffic which remains un-classified as P2P (in the first step) is then fed to the flow-level classification process where flow-heuristics are combined with the statistical-based technique to perform a classification of the remaining traffic. The proposed technique is implemented in Java with the help of the jNetPcap library [56] and Weka [57].

Figure 2. Multi-level P2P traffic classification technique.

While performing the task of traffic classification, a combination of five network parameters (i.e., source-IP, destination-IP, source-port destination-port, and protocol) are generally used to define the traffic-flow [36]. All the communication that happens among the two processes will share these same five parameters. In the packet-level classification process, packets belonging to the same flow are recognized by calculating the hash-key of the packet through combining the five-tuple flow information, as shown in Figure 3. In this way, packets belonging to the same flow and travelling in either direction will have the same hash-key. This hash-key is useful to find out if the packets belonging to the flow have already been classified as P2P or not.
We use the P2P flow table to store the flow-details of those flows, which are already classified as P2P. The information stored in this table will be used to verify whether a particular traffic flow (under analysis) is already classified earlier as P2P flow or not. Moreover, we use a separate table, namely the P2P destination-IP-table, to store destination < IP, port > pair information of those flows, which are already classified as P2P. This information is useful in the heuristic-based classification process.

4.2. Packet-Level Classification Process (First Step)

As shown in Figure 2, initially a pre-processor is used which captures the network traffic and filters out unwanted packets to create the traffic dataset. The traffic is then fed into the packet-level classification process, which is illustrated in Figure 4. Here, the packet-level classification process combines the P2P-port based technique and packet-heuristic based technique for classifying P2P traffic. In this level, as the network packet arrives for processing, its hash-key (as shown in Figure 3) is calculated and mapped with the information stored in the P2P flow table (which contains the records of the already classified P2P flows) in order to verify whether the traffic flow of that packet is already classified as P2P flow or not. If a match is found, then the new packets are fetched and this step is repeated (as shown in Figure 4).

![Figure 3. Calculation of the packet hash-key.](image)

**Figure 3.** Calculation of the packet hash-key.

![Figure 4. Packet-level classification process (first step).](image)

**Figure 4.** Packet-level classification process (first step).

4.2.1. P2P-Port Based Classification

The packets are initially fed to the P2P-port based classification technique, where the TCP/UDP port number is extracted from the packet header and mapped with the database of well-known P2P port numbers [6,7] for communication.
port numbers (shown in Table 1 in the previous section) used by various P2P applications. If a match is found, then its flow is classified as P2P. Accordingly, the flow-details are added in the P2P flow table and the destination-IP-table is updated. Although it is known that the port-based technique is inefficient in traffic classification, it has been used here for the purpose of performing an early classification of the P2P traffic, which may not be masquerading and still using well-known P2P port numbers [6,7] for communication.

4.2.2. Packet-Heuristic Based Classification

The traffic which remains unclassified as P2P is fed to the packet-heuristic based classification technique. Here, the traffic flows are classified as P2P or non-P2P on the basis of the proposed heuristic rules. If a traffic flow satisfies any of the proposed heuristics, then it is classified as P2P flow, and this information is updated in the P2P flow table and destination-IP table, accordingly. The heuristics used in the proposed technique for classifying P2P traffic are discussed below:

1. **Usage of ephemeral port numbers**: In order to communicate over a network, an application makes use of the transport-layer port number. The port numbers below 1024 are called well-known privileged port numbers, whereas port numbers above 1024 are called ephemeral port numbers. It is observed that many P2P applications (e.g., BitTorrent, VoIP, etc.) use ephemeral port numbers, whereas non-P2P applications (e.g., web, email, etc.) use well-known privileged port numbers for communication over the network. In client-server-based communication, the client uses an ephemeral port number (randomly chosen by the operating system) to communicate with the server and the server responds back with the requested data using a well-known port number. Therefore, if the source port and destination port of a packet are found to be ephemeral, then its flow is classified as P2P. However, this heuristic fails if a peer masquerades using the well-known port number (e.g., port 443 used by HTTPS) for communication.

2. **Usage of TCP and UDP protocols simultaneously**: It has been observed that most of the P2P applications such as Skype, Gnutella, etc. employ TCP and UDP protocols simultaneously for communication. Depending on the type of P2P application, TCP may be used for transferring the data, whereas UDP may be used for signaling messages and vice-versa [12,58]. For example, a Skype peer communicates with the super-peer using both TCP and UDP protocols. Therefore, if a source-IP uses TCP and UDP protocols simultaneously for communication with the destination-IP, then its flow is classified as P2P. However, some false positives may exist with this heuristic as there are some non-P2P applications such as streaming, IRC, gaming, etc. which exhibit a similar behavior [5].

3. **Communication with destination-IP which is already classified as P2P**: Prior to the communication between peers, a peer waits for the incoming connections from the other peers with the help of a listening port [59]. Figure 5 shows a scenario where peer-A (already classified as P2P) waits for incoming connections from the other peers. Its < IP, port > pair will act as the destination for all the other peers (i.e., peer-B, peer-C, peer-D, etc.) who want to communicate with it. Hence, the flows of all such peers are classified as P2P which communicate with the already classified P2P peer. For this purpose, we make use of the P2P destination-IP-table for storing < IP, port > pair information of those peers, which are already classified as P2P. While processing the packets, we analyze if either their source or destination < IP, port > pair maps with one of the records stored in the destination-IP-table, then the flows of such packets are also classified as P2P.

4. **Usage of consecutive port numbers**: It has been observed that various P2P applications actively make a number of connections with the other peers for communication. In this case, the operating system of a peer allocates successive port numbers to the application (where the first port is randomly chosen and allocated) [60]. Figure 6 shows a scenario where the P2P source peer-A uses consecutive port numbers to communicate with the destination peers (i.e., peer-B, peer-C, peer-D, etc.). Therefore, we analyze that if a source-IP makes use of consecutive port numbers for communication, then its flows are classified as P2P.
As various P2P applications communicate either via TCP or UDP (or both), it has been analyzed that the proposed heuristic rules work equally with both TCP and UDP traffic and are not affected even if the traffic is encrypted. Algorithm 1 shows the packet-level classification process which classifies P2P traffic as the first step. The traffic which remains un-classified as P2P is fed to the flow-level classification process (i.e., second step).

**Figure 5.** Connection pattern of source peers with the destination P2P peer.

**Figure 6.** Connection pattern of source P2P peer with the destination peers.
Algorithm 1: Packet-level classification process (first step).

Input: Network traffic packets  
Output: Traffic-flows classified as P2P and non-P2P

pkt: Packet  
ft: P2P_flow_table  
fi: Flow_information  
spn: Source_port_number  
dpn: Destination_port_number  
dit: Destination_IP_table  
wkP: Well_known_P2P_ports  
h1: Heuristic_1  
h2: Heuristic_2  
h3: Heuristic_3  
h4: Heuristic_4

Begin
(1) pkt = fetch_packet()
(2) do
(3) {  
(4) if (ft.contains (pkt.fi))
(5) goto step 15
(6) else if (pkt.spn == wkP || pkt.dpn == wkP)
(7) {  
(8) write: pkt.fi → P2P
(9) update: dit ← pkt.fi
(10) }
(11) else if ((pkt.h1 || pkt.h2 || pkt.h3 || pkt.h4) == true)
(12) write: pkt.fi → P2P
(13) else
(14) write: pkt.fi → non-P2P
(15) pkt = fetch_packet()
(16) } while (pkt != NULL)
(17) goto 2nd step classification process  
End

4.3. Flow-Level Classification Process (Second Step)

Figure 7 shows the flow-level classification process, which combines the flow-heuristic based technique and statistical-based technique (using C4.5 decision tree). The traffic which remains unclassified as P2P in the packet-level classification process is fed to the flow-level classification process. Here, initially before processing a traffic-flow, its information is searched in the P2P flow table (which contains the records of the already classified P2P flows) in order to verify whether it is already classified as P2P flow or not. The flows which are not classified as P2P are fed to the flow-heuristic based classification process which is explained below.
4.3.1. Flow-Heuristic Based Classification

One of the properties of the P2P application is that it acts as both a client and server at the same time, i.e., data are transferred from destination-to-source and source-to-destination simultaneously. A similar behavior can be detected in client-server applications as well, where data are transferred from the client to a server with request messages and the server responds with the requested data. However, the main difference is that the amount of data sent from the client to a server (i.e., request messages) is very small compared to the amount of data sent from the server to a client (i.e., data requested). However, in the case of the P2P application, the data are sent in both directions (i.e., from source-to-destination and destination-to-source) in a large amount. Therefore, we analyze that if in a flow, the amount of data sent in each direction (i.e., destination-to-source and source-to-destination) is greater than the threshold-value, then the flow is classified as P2P. For experimental purposes, the threshold-value taken here is 3 MB.

4.3.2. Statistical Based Classification

The traffic-flows which still remain unclassified as P2P (in the previous process) are fed to the statistical-based classification process, where statistical features of the traffic-flows are extracted and used with the C4.5 ML algorithm to classify the remaining traffic (as shown in Figure 7). This process involves the training phase as well as the classification phase. In the training phase, a classification model is built using the training dataset which contains both P2P and non-P2P traffic-flows. The ML algorithm analyzes the relationship between the flow features and the output class value to generate a classifier model, which predicts the type of traffic flow by analysing its statistical features. In the classification phase, statistical features of a traffic flow are extracted and fed into the classifier model. If the characteristics of a flow matches the distinct characteristics of P2P traffic, then the flow is classified as P2P.

Traffic-flow features are the numeric values calculated over numerous packets belonging to that flow. The flow features which are used with the ML algorithm in the proposed technique are mentioned below:
• Packet inter-arrival time from source-to-destination
• Packet inter-arrival time from destination-to-source
• Duration of flow
• Total number of packets from source-to-destination
• Total number of packets from destination-to-source
• Total number of bytes of all packets
• Total packet bytes from source-to-destination
• Total packets bytes from destination-to-source
• Payload size of packets from source-to-destination
• Payload size of packets from destination-to-source

These flow features have been mostly used in previous studies [20], as well. They are given as input to the ML algorithm to build a statistical-based classifier for performing the classification. The C4.5 ML algorithm is chosen for traffic classification purposes, since it is faster and better compared to other ML algorithms [61]. Algorithm 2 shows the flow-level classification process (i.e., second step) which classifies P2P traffic at a flow level.

**Algorithm 2:** Flow-level classification process (second step).

**Input:** Traffic-flows classified as non-P2P in the first step

**Output:** Traffic-flows classified as P2P and non-P2P

flw: Flow
tf: P2P_flow_table
fi: Flow_information
std: Data_transferred_from_source_to_destination
dts: Data_transferred_from_destination_to_source
thld: Data_threshold (3MB)
fh: Flow_heuristic = (std + dts) > thld
ff: Flow_features
MLA: Machine_learning_algorithm
rst: Result

Begin
(1) flw = fetch_flow()
(2) do
(3) |
(4) if (tf.contains (flw.fi))
(5) goto step 17
(6) else if (flw.fh == true)
(7) write: flw → P2P
(8) else
(9) |
(10) fset = flw.ff
(11) rst = flw.MLA (fset)
(12) if (rst == “P2P”)  
(13) write: flw → P2P
(14) else
(15) write: flw → non-P2P
(16) }  
(17) flw = fetch_flow()
(18) }while (flw ! = NULL)
End
5. Verification

5.1. Evaluation Metrics

The performance of a classifier can be characterized using the metrics known as: False positive (FP), false negative (FN), true positive (TP), and true negative (TN). They are described as follows:

1. TP: Percentage of instances correctly categorized as belonging to a particular class.
2. TN: Percentage of instances correctly categorized as not belonging to a particular class.
3. False positive (FP): Percentage of instances incorrectly categorized as belonging to a particular class.
4. FN: Percentage of instances incorrectly categorized as not belonging to a particular class.

The proposed technique classifies the traffic flow as P2P or non-P2P. Accuracy (1), recall (2), and precision (3) metrics are used to evaluate the proposed methodology. Accuracy is used to measure the capability of the classifier for identifying negative and positive cases. Recall is used to measure the overall percentage of correctly classified cases. Precision is used to measure the percentage of correctly classified positive cases. They are defined as follows:

\[
\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)
\]

\[
\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (2)
\]

\[
\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (3)
\]

5.2. Datasets, Validation, and Experimental Results

To evaluate the proposed technique, two offline traffic datasets have been used, which are realistic, and consist of both P2P and non-P2P flows as shown in Table 2.

Table 2. The number of flows in the datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>P2P (No. of Flows)</th>
<th>Non-P2P (No. of Flows)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset-1</td>
<td>20,617</td>
<td>48,179</td>
<td>68,796</td>
</tr>
<tr>
<td>Dataset-2</td>
<td>3881</td>
<td>2892</td>
<td>6773</td>
</tr>
</tbody>
</table>

The first traffic dataset (i.e., Dataset-1) is UNIBS [62,63] which belongs to the University of Brescia and the second traffic dataset (i.e., Dataset-2) is collected at the campus area network in a controlled environment using the Wireshark [64] tool and their pattern of communication was observed. Therefore, the flows which belong to the P2P traffic are well-known in advance. In addition, such traffic flows are labelled accordingly with actual applications for the purpose of ground-truth verification, which consist of traffic traces of different application protocols, for example, HTTP, SMTP, BitTorrent, Skype, Dropbox, DNS, FTP, POP3, IMAP, etc., as shown in Table 3.

Table 3. Summary of the collected data.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Packets</th>
<th>Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP3</td>
<td>13,647</td>
<td>918,878</td>
</tr>
<tr>
<td>IMAP</td>
<td>3191</td>
<td>213,554</td>
</tr>
<tr>
<td>HTTP</td>
<td>1,399,230</td>
<td>92,060,704</td>
</tr>
<tr>
<td>BitTorrent</td>
<td>379,836</td>
<td>329,477,265</td>
</tr>
</tbody>
</table>
We made the training and testing dataset by combining both datasets, as shown in Table 2. In the statistical-based classification process, the datasets were divided into training and testing parts using the k-fold cross-validation procedure. Nowadays, most of the communication between the peers over the network is encrypted to provide security or to obfuscate the traffic. Therefore, for experimental purposes, Dataset-2 was constructed with the encrypted P2P traffic to test the classification performance of the proposed hybrid technique. The results show that the proposed hybrid technique achieves overall accuracy, recall, and precision values ranging between 97.4–98.3%, 97.9–98.4%, and 95.9–97.6%, respectively (as shown in Figure 8), which also show that it is able to classify the encrypted traffic. Figures 9 and 10 show that the classification accuracy achieved by the proposed hybrid technique is higher than the various existing hybrid, as well as non-hybrid P2P traffic classification techniques.
In the proposed hybrid technique, after analyzing the type of protocol (i.e., TCP or UDP) used by packets for communication, either TCP or UDP port numbers are extracted from packet headers to perform the P2P-port based classification. In both packet-heuristic and flow-heuristic based classifications, the proposed heuristic rules analyze behavior/communication patterns of traffic, which are not affected whether a flow uses TCP or UDP protocol for communication. At last, the statistical-based classifier uses various statistical features of traffic (with C4.5 decision tree) to perform the classification, which are independent of traffic using the TCP or UDP protocol for communication. Hence, the overall proposed hybrid technique is able to work with both TCP and UDP protocols at every step. In addition, it also involves less computation, since it does not rely on the DPI technique (which requires a large amount of computation for inspecting the traffic) to perform the classification, but rather relies on heuristic-based and statistical-based techniques which are comparatively light on resources [6].

Furthermore, the classification performance of the proposed hybrid technique at various stages is shown in Table 4. It can be seen that the packet-level process (which is a combination of P2P port-based and heuristic-based techniques) achieves an accuracy of 90.50% in classifying P2P traffic. When it is combined with the flow-level process (which is a combination of flow-heuristic and statistical-based techniques), then the classification accuracy reaches 98.30%. This can be attributed to the fact that some P2P applications use masquerading techniques or hide their traffic behind well-known port numbers (which could not be classified in the packet-level process) and hence such traffic is classified using the flow-level classification process.

Table 4. Classification performance at various steps (P → P2P-port-based, PH → packet-heuristic-based, FH → flow-heuristic-based, S → statistical-based).

<table>
<thead>
<tr>
<th>Classification Process</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>11.90</td>
</tr>
<tr>
<td>P + PH</td>
<td>90.50</td>
</tr>
<tr>
<td>P + PH + FH</td>
<td>95.10</td>
</tr>
<tr>
<td>P + PH + FH + S</td>
<td>98.30</td>
</tr>
</tbody>
</table>

In Table 4, it can be seen that although the P2P-port based technique (i.e., P) is inefficient in classification, it has been utilized here since it is the fastest method to classify traffic if it does not masquerade and use well-known P2P port numbers [6,7] for communication. Therefore,
its main purpose is to reduce the amount of traffic that needs to be analyzed by heuristic-based techniques (i.e., PH and FH) if it classifies some P2P traffic at an early stage. The advantage of using the heuristic-based technique (i.e., PH and FH) is that it classifies traffic based on its behavior/communication pattern and does not require much computation for the analysis compared to DPI statistical-based techniques. Finally, the advantage of using the statistical-based classifier (i.e., S) in the proposed technique is that it classifies any remaining P2P traffic which could not be identified by heuristics (i.e., PH and FH), where such P2P traffic may escape detection (from heuristics) using some masquerading technique, or may belong to an application which is newly emerged and has an entirely different (or new) communication pattern. However, as the statistical-based classifier performs the classification on the basis of various statistical features of traffic, therefore, its limitation is that the model needs to be trained (and updated accordingly) to identify new applications (which require some time). For example, a new P2P application with a communication pattern similar to existing P2P applications but having different traffic statistics may be classified incorrectly by the classification-model until it is re-trained.

Table 5 shows the summary regarding the approach used in the first and second step classification process along with the classification accuracy of various existing hybrid P2P traffic classification techniques.

<table>
<thead>
<tr>
<th>Ref</th>
<th>Studies</th>
<th>First Step Classification</th>
<th>Second Step Classification</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[53]</td>
<td>Chen et al. (2009)</td>
<td>Static feature-based (port, signature, pattern)</td>
<td>Flexible neural tree-based</td>
<td>95.65</td>
</tr>
<tr>
<td>[52]</td>
<td>Li et al. (2009)</td>
<td>Coarse-grained (C4.5 decision tree)</td>
<td>Fine-grained (signature, port)</td>
<td>96.03</td>
</tr>
<tr>
<td>[55]</td>
<td>Sajeev and Nair (2016)</td>
<td>Signatures</td>
<td>C4.5 decision tree</td>
<td>96.00</td>
</tr>
<tr>
<td>[22]</td>
<td>Khan et al. (2019)</td>
<td>Port filtering, DNS query filtering, flow counting</td>
<td>Decision tree</td>
<td>94.40</td>
</tr>
</tbody>
</table>

| Proposed hybrid technique | port, packet-level heuristics | Flow-level heuristics, C4.5 decision tree | 98.30 |

During the classification process, the techniques used in [5,19–21,52,55], rely on the signature-based approach, which is computationally expensive [6,30,31] and has various other limitations as discussed in Section 2. In addition, the techniques in [19–21] do not classify the UDP traffic. The technique used in [53] relies on dedicated hardware for the P2P classification, whereas the technique used in [22] may lead to many false negatives since during the classification process, it filters out all the traffic using well-known port numbers (such as 20, 21, 443, etc.) by considering them as non-P2P traffic. The hybrid technique proposed in this paper not only achieves high P2P classification accuracy, but also involves less computation since unlike existing various hybrid techniques (mentioned above), it does not rely on the signature-based technique which is computationally expensive and unsuitable for high-speed networks [6,30,31], but rather relies on heuristic-based and statistical-based techniques, which are comparatively light on resources. In addition, the proposed hybrid technique works with both TCP and UDP traffic flows and classifies the encrypted traffic, as well.
6. Conclusions

P2P applications have been widely used since the past decade and bring a lot of conveniences, but pose various issues to the ISPs and enterprises in the tasks related to providing QoS for various applications, addressing network congestion, security, etc. Conventional techniques for traffic classification such as port-based and payload-based are ineffective in classifying P2P traffic due to various limitations associated with them. Therefore, modern techniques need to be adopted for classifying P2P traffic with high accuracy, which will allow ISPs or network administrators to either limit or ban P2P traffic in order to maintain a Quality of Service for various applications in their network.

In this work, we propose the multi-level P2P traffic classification technique which is sub-divided into the packet-level and flow-level classification process. By analyzing the behavior of various P2P applications, some heuristic rules have been proposed to classify P2P traffic and are utilized in both the packet-level and flow-level classification process. If the traffic remains unclassified as P2P, then it undergoes further analysis using statistical-features of traffic which are used with the C4.5 decision tree to classify traffic as P2P or non-P2P. The experiments performed using the proposed hybrid technique achieved high classification accuracy of 98.30%, which is higher than other hybrid/non-hybrid techniques as it combines the benefits of both heuristic-based (less computation as compared to DPI) as well as statistical-based (scalability) techniques. In addition, it also works with both TCP and UDP traffic and is not affected even if the traffic is encrypted. However, there are certain limitations of the proposed technique:

1. It may produce some false positives (during the P2P-port based classification process) if the network traffic includes malicious applications using well-known P2P default ports that can be utilized by various P2P applications.
2. It does not perform a fine-grained classification to classify P2P traffic into specific ports.
3. It is made to work on offline datasets which consist of a limited number of P2P and non-P2P applications and hence, may produce some false positives (due to one of the proposed heuristics discussed in Section 3) when traffic datasets involve all kinds of P2P and non-P2P application protocols.

Hence, in the future, we plan to enhance the proposed technique, which can perform a fine-grained P2P classification (i.e., identify P2P application traffic specifically), as well. In addition, broader traffic datasets (containing traffic traces of various popular P2P and non-P2P applications) would be used for analyzing the effectiveness of the technique. 

Author Contributions: Conceptualization, M.B. and V.S.; methodology, M.B. and V.S.; software, M.B.; validation, P.S. and M.M.; formal analysis, M.B. and P.S.; investigation, V.S. and M.M.; resources, M.B.; data curation, M.B. and V.S.; writing—original draft preparation, M.B., P.S. and M.M.; writing—review and editing, M.B., V.S. and M.M.; visualization, M.B. and P.S.; supervision, V.S. and M.M.; project administration, P.S. and M.M.; funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: Taif University Researchers Supporting Project number (TURSP-2020/10), Taif University, Taif, Saudi Arabia.

Acknowledgments: We are very thankful for the support from Taif University Researchers Supporting Project (TURSP-2020/10).

Conflicts of Interest: The authors declare no conflict of interest.

References


30. Liu, S.-M.; Sun, Z. Active learning for P2P traffic identification. Peer-to-Peer Netw. Appl. 2015, 8, 733–740. [CrossRef]


36. Sun, M.-F.; Chen, J.-T. Research of the traffic characteristics for the real time online traffic classification. J. China Univ. Posts Telecommun. 2011, 18, 92–98. [CrossRef]


Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.