BEHAVIORAL ANALYSIS OF PEER-TO-PEER NETWORK TRAFFIC USING STATISTICAL TECHNIQUE TO IMPROVE IDENTIFICATION ACCURACY

A Thesis

Submitted for the award of the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE AND ENGINEERING

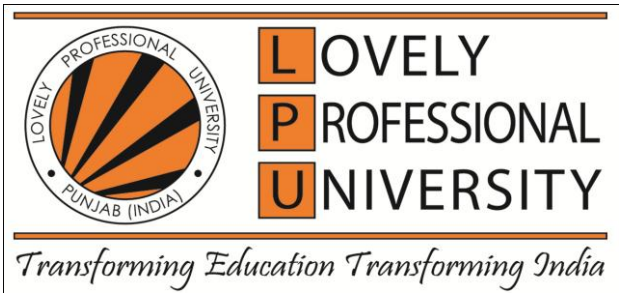
By

Max Bhatia

Registration no. 41400131

Supervised By

Dr. Vikrant Sharma



LOVELY PROFESSIONAL UNIVERSITY

PUNJAB

2021

DECLARATION

I hereby declare that the research work reported in the thesis entitled **"BEHAVIORAL ANALYSIS OF PEER-TO-PEER NETWORK TRAFFIC USING STATISTICAL TECHNIQUE TO IMPROVE IDENTIFICATION ACCURACY"** in partial fulfillment of the requirement for the award of Degree of Doctor of Philosophy in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor **Dr. Vikrant Sharma**. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University's Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this thesis represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my thesis work.

Signature of Candidate

Max Bhatia

Registration no. 41400131

CERTIFICATE

Certified that **Max Bhatia** (Registration no. 41400131) has carried out the research work presented in this thesis entitled **"BEHAVIORAL ANALYSIS OF PEER-TO-PEER NETWORK TRAFFIC USING STATISTICAL TECHNIQUE TO IMPROVE IDENTIFICATION ACCURACY"** for the award of Degree of Doctor of Philosophy in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab**,** under my supervision. The thesis embodies results of original work, and are carried out by the student himself and the content of the thesis do not form the basis for the award of any other degree to the candidate or anybody else from this or any other University/Institution.

Signature of Supervisor

(Dr. Vikrant Sharma)

Date:

ABSTRACT

Peer-to-Peer (P2P) architecture consists of distributed systems interconnected in such a way that each participating peer can share its resources such as files, storage capacity, processing power, etc., to the other peers over the network without requiring a centralized server. By utilizing these resources, various P2P services are implemented, such as file-sharing, audio/video streaming, online gaming, etc., which are accessible by all the peers of a network.

P2P applications have been popular among users for more than a decade. They consume a lot of network bandwidth, due to which Internet Service Providers (ISPs) & network administrators face various network challenges such as congestion, security, managing resources, etc. Hence, its accurate classification will allow them to address multiple network-related tasks such as network bandwidth planning, policy-based traffic management, fault diagnosis, Quality of Service (QoS) analysis for applications, etc. Hence, there arises a need to monitor and classify the internet traffic generated by P2P applications. This field is actively researched as new application protocols keep on emerging. Nowadays, classifying P2P traffic with high accuracy is difficult since various P2P applications either masquerade or encrypt their traffic to avoid detection.

When the internet began, network traffic classification was an easy task that could be achieved using a simple and easy-to-implement approach called the port-based technique. This technique could easily classify legacy applications like HTTP, FTP, DNS, etc., with high accuracy, but it soon became inefficient/inaccurate in traffic classification since P2P applications started using random port numbers for communication to navigate through firewall & other network restrictions. Therefore, another approach called the payload-based technique was adopted, which classifies network traffic by inspecting its packet payload. Such an approach has high accuracy in classifying network traffic. Still, despite this fact, it cannot be applied in various situations such as unavailability of payload information, traffic payload encryption, etc. Therefore 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. Due to the limitations of conventional techniques, the modern classification approach called Classification in the Dark is adopted, which classifies P2P traffic either by using statistical features of traffic flows or observing behavioral patterns associated

with the traffic flow. But, the accuracy of this technique relies heavily on the robustness of statistical features or behavioral patterns selected for classification.

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 employed in this research work, which utilizes the benefits of both heuristic and statistical-based techniques.

In the research work, initially, we focused on classifying P2P traffic in the network by analyzing the behavior of various P2P applications. We proposed a multi-level classification technique which is a combination of heuristic-based & statistical-based techniques. In the heuristic-based classification technique, heuristic rules have been proposed for classifying P2P network traffic. The traffic which remains unclassified as P2P undergoes further analysis where statistical-based classification technique is employed on the statistical features of the traffic to classify the traffic either as P2P or non-P2P.

Further, the research work concentrates on classifying network traffic generated by various P2P file-download applications such as uTorrent, eMule, etc. For this purpose, a 2-step traffic classification technique is proposed, combining heuristic-based and statistical-based techniques. We identified a set of heuristic rules and unique packet size distribution of P2P file-sharing traffic with the help of real offline traffic traces to classify P2P-file-sharing (P2P-fs) traffic. The traffic which remains unclassified as P2P-fs undergoes further analysis where statistical-based classification technique is employed on the statistical features of the traffic to classify the traffic either as P2P-fs or non-P2P-fs.

At last, the research work focused on classifying network traffic generated by various P2P-VoIP applications such as Skype, Google-meet, etc. Here, we specifically focus on classifying video traffic generated by P2P-VoIP applications. For this purpose, a 2-step traffic classification technique is proposed by combining heuristic-based and statistical-based techniques. We identified a set of heuristic rules and unique packet size distribution of VoIP (video) traffic with the help of real offline traffic traces to classify VoIP traffic. The traffic that remains unclassified as P2P-VoIP undergoes further analysis in the statistical-based technique (with the machine learning algorithm, namely C4.5 decision tree), which utilizes traffic's statistical features to classify it as VoIP or non-VoIP.

ACKNOWLEDGEMENT

Firstly, I would like to thanks my guide Dr. Vikrant Sharma. His encouragement, enthusiasm, and support motivated me to pursue this research work. I offer my gratitude to him for all of his time and energy, which he spent for me, by discussing everything regarding research, career choices, reading my papers, and providing proper guidance in my research work through various obstacles. His professional and caring approach towards all the people he works with has genuinely inspired me.

I owe my thanks to the faculty members of the department for their valuable feedback. I would like to thank all my friends for their direct and indirect help and support.

I would especially like to thank my family members wholeheartedly since I would not be anywhere in my life without their help and support. The love, patience, and happiness I received from them helped me achieve the stage of my life where I am currently.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Full Form
ADSL	Asymmetric Digital Subscriber Line
DNS	Domain Name System
DPI	Deep Packet Inspection
EMEA	Europe, the Middle East, and Africa
FN	False Negative
FP	False Positive
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ISP	Internet Service Provider
LAN	Local Area Network
ML	Machine Learning
NAT	Network Address Translation
P2P	Peer-to-Peer
P2P-fs	Peer-to-Peer-file-sharing
psd_ratio	packet-size-distribution_ratio
PSTN	Public Switched Telephone Network
QoS	Quality of Service
SMTP	Simple Mail Transfer Protocol
TN	True Negative
TP	True Positive
VoIP	Voice over Internet Protocol

CHAPTER 1

INTRODUCTION

1.1 Introduction

Peer-to-Peer (P2P) architecture consists of distributed systems interconnected to each other, which forms a dynamic overlay network. The participating computer systems are called "peers," which have the ability to share their resources such as files, storage capacity, processing power, etc., over the network without requiring a centralized server. These shared resources can be used to implement various services over the network, such as file-sharing, audio/video streaming, online gaming, etc., which are accessible by all the peers of the network. P2P architecture is different from client-server architecture. A centralized server is responsible for providing the resources to its clients in a client-server network, where clients request the resource from the server, and the server responds. On the other hand, in a P2P network, every peer acts as a client & server simultaneously, thereby contributing/requesting the resources to/from the other peers at the same time. Architectural difference between client-server network and P2P network has been depicted in Figure 1.1 & Figure 1.2, respectively.

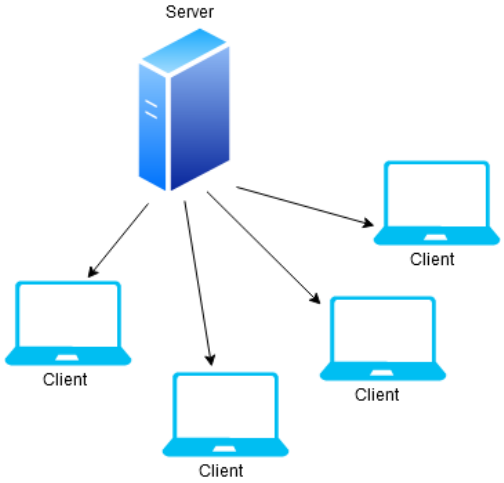


Figure 1.1. Client-server architecture.

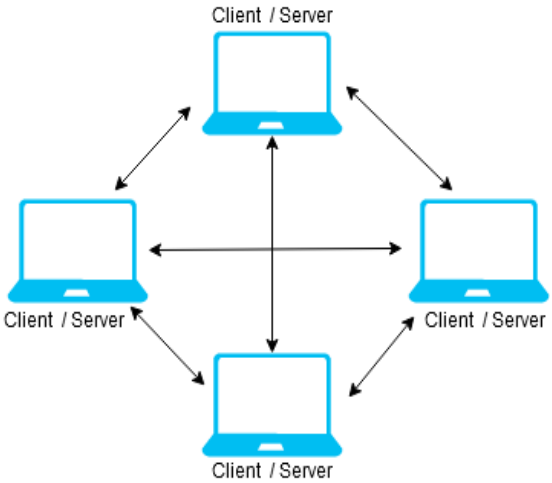


Figure 1.2. P2P architecture.

In a P2P network, peers establish a direct connection with each other to share various services and resources and do not require a centralized server for communication. Hence, such a network can be viewed as a pool of shared resources where every participating peer requests/provides the resources to each other. So generally, two kinds of traffic can be seen on the internet: client-server traffic and P2P traffic.

1.2 Internet traffic and its measurement

In the past, Internet traffic depended on the client-server model where the client used to request the data, and the server provided it, leading to asymmetric network traffic. With the evolution of the Internet, internet peers got the privilege to distribute their data which could be shared with other peers on the Internet. Further, P2P traffic started evolving at the beginning of the 21st century, which incorporated direct dissemination of data between peers on the Internet. In such a scenario, peers started behaving as a client & server simultaneously, thus downloading the contents, they required from other peers and distributing their contents to other peers. Due to this, network traffic has become symmetric. From the network management point of view, P2P traffic needs to be identified as it involves traffic flowing in both directions at the same time, thus consuming more bandwidth. In this system, peers share the distribution cost of the service instead of relying on a dedicated server for it. This is advantageous for the service providers for distributing the contents, but only at the cost of producing more traffic in the network. There is an increase in the number of communications between the peers for searching the content from the remote peers, which has resulted in a large number of connections as compared to the client-server system where only a few connections were formed. Thus, P2P systems produce a large amount of traffic as opposed to client-server systems. This poses an issue where network traffic needs to be monitored and controlled so that P2P traffic alone doesn’t consume a large portion of the available bandwidth. Hence, a balance needs to be maintained so that various other traffic such as HTTP, FTP, SMTP, etc., also get their fair proportion of network bandwidth. It ensures that Internet Service Providers (ISPs) can provide Quality of Service (QoS) to every application by implementing specific policies. Further, conventional devices are unable to control P2P traffic effectively due to which ISPs face several other challenges such as paying for added traffic requirements, satisfying customers with excellent experience of broadband service, purchasing backbone links & upstream bandwidth which are costly.

Internet traffic has been growing rapidly over the past few years [1]. This is attributed to the fact that P2P traffic has grown at such a pace that various types of applications have been emerging over time. Various application protocols such as HTTP, SMTP, etc., no longer dominate Internet traffic which has instead been taken over by P2P traffic to a large extent [2]. P2P file-sharing has been a significant trend in recent years. The major content which is shared or distributed through P2P applications is audio, video, and games which tend to be large in terms of file size [3]. This also includes illegal file sharing. P2P traffic is one of the largest contributors to internet traffic [4], which consumes a significant chunk of network bandwidth. Azzouna and Guillemin [5], in their study, identified that 49% of traffic was due to P2P applications in the link of Asymmetric Digital Subscriber Line (ADSL). A worldwide study conducted by ipoque [6] (in 2008) about Internet traffic displayed that P2P file-sharing applications produce a large amount of traffic compared to various applications taken together. Therefore, identifying the application that produces traffic becomes crucial to accomplish the tasks such as implementing billing mechanisms, maintaining Quality of Service for applications, implementing security measures, etc. Now it is a very difficult task as there are umpteen issues associated with it.

The traditional method used to accomplish the task of network traffic classification includes associating port-numbers of transport-layer to the well-known application protocols. But this technique of identifying applications soon became ineffective as numerous applications started transferring their data using random port numbers. Also, various applications used masquerading techniques by utilizing well-known port numbers (e.g., 80 utilized by HTTP) to hide their traffic. Karagiannis et al. [7] identified that many P2P applications utilize port number 80 to transfer their data and also found that 30 to 70% of the P2P traffic utilized random port numbers. Madhukar and Williamson [8], in their study, showed that Internet traffic could not be identified correctly by using port-based methods. Due to these issues, another technique based on payload inspection was adopted. Although this technique proved to be of great accuracy, it also possessed various limitations such as the requirement of a large amount of computational resources, privacy issues involved, and the inability of this technique to work when the payload is encrypted. Hence, another alternative to identify traffic was adopted based on statistical or behavioral methods such as packet size, total packets sent, total packets received, etc., which do not possess limitations posed by port-based or payload-based techniques.

Williamson, in [9], considering the study of the network, classified the research tools for as Online & Offline, LAN & WAN, Active & Passive, Protocol level, and Hardware & Software. The significance of every category depends upon the research purpose. Their brief description for traffic classification is given below:

- **Online and offline:** Online approach involves analyzing traffic while it is currently flowing through the network. Such a process requires high computational power and resources in high-speed networks but is greatly useful in applications such as in NIDSs and firewalls when instant decisions or actions are required to be made for the packets currently flowing in the network. In contrast, the Offline approach involves network traces to be collected as an offline file for analyzing at a later time when the packets have already crossed the network. This approach is mostly preferred when a real-time analysis is not required, and it is also useful for research and validation, as one can run several approaches on the same set of traces which can be compared for results.
- **LAN and WAN:** Measurements conducted for traffic classification purpose is preferably done on LAN instead of WAN since the former involves no loss of information whereas latter one is difficult to get access to.
- **Hardware and software:** Dedicated hardware tends to give better solutions in terms of performance which are useful in real-time analysis. For traffic measurement, monitoring, or capturing, some companies like Endace [10], ipoque [11], Wildpackets [12], and Napatech [13] provide hardware-based solutions. As researchers are primarily interested in analyzing Ethernet frames or IP packets during traffic classification, it is of less significance whether the analysis is done using a hardware-based or software-based solution.
- **Protocol level:** The researchers can achieve internet traffic measurement at various (or even multiple) protocol levels, but for traffic classification purposes, the researchers primarily consider Ethernet level or IP level.
- **Active and passive:** Active approach involves analyzing the traffic behavior by injecting actual packets into the network. It allows one to control the simulation scenario, such as the type of traffic flowing in the network, its frequency, etc. But its limitation is that it puts extra load on the network bandwidth and can affect the performance of routers or switches. Also, the actual behavior of traffic flow is not indeed reproduced by this approach, which may affect the results. On the other hand, the Passive approach doesn’t need to inject any

packets into the network and captures and analyses the actual traffic flowing through the network. Hence, it doesn't affect the performance of bandwidth or any network equipment, and measurements made using this approach reflect the actual behavior or properties of real traffic. But its limitation is that it produces an enormous amount of data that needs to be handled and analyzed to obtain useful information.

For traffic identification or classification purpose, the researchers mostly focus on IP packets or Ethernet frames. In the Per-packet approach, each packet traveling in the network is captured to analyze the traffic. It can be useful in certain scenarios such as Network Intrusion Detection Systems (using tools like Snort [14], Bro [15]) where some decisions need to be made on each packet traveling through the network. Also, these packets can be captured and stored for offline analysis by using tools such as Wireshark [16] and Ettercap [17]. They can mine necessary information from the layers of the protocol stack by inspecting each packet. Although packets flowing through the network are individual data units, there exist certain relationships between them, such as packets generated by the same request or response, packets belonging to the same application containing data, etc., and hence such hidden information can be mined by using Per-Flow analysis. A flow is mostly defined as the set of packets sharing common characteristics: Source-IP, Destination-IP, Source-Port, Destination-Port, and Protocol [18]. The flow is considered active when the time interval between packets (of to a particular flow) is below a certain threshold value, which depends on the purpose of the analysis or study. Claffy et al. [19] identified that a threshold value of 64s is a good bargain considering the flow size and initializing & terminating flows. Also, a flow can be defined as unidirectional if there is no discrimination between packets traveling in either direction. Hence, it is considered a single flow; or it can be defined as bidirectional if one considers packets flowing in either direction separately as two separate flows. Unidirectional flows are useful in studies such as managing network bandwidth management and measuring the performance of a network, where there is a need to find the disparity in traffic traveling in either direction. In contrast, bidirectional flows are considered helpful in scenarios such as analyzing the sessions of TCP connection. Also, this approach is more appropriate for traffic classification where traffic flowing between two sides is produced by the same application and is associated with the same class. There are some tools available for performing flow-based analysis, such as Coral-Reef [20], to analyze the packets from the network adaptors or offline traffic traces. Tools such as Cisco Netflow [21] can directly obtain information about the traffic flow from the router and other elements of the network.

1.3 Traffic data collection and trace reduction

Traffic data collection in a network should be done with care to protect users' privacy and other data containing sensitive information. Some of the good practices and considerations have been mentioned in [22]. In the Passive approach, traffic flows can be gathered from the routers by using protocols namely IPFIX, or the trace files can be generated by capturing packets with the help of software like tcpdump [23], WinDump (Windows version) [24], or other available tools which are based on libpcap [23] or WinPcap [24] libraries. But, using such techniques results in the generation of large trace files, which require more processing power and storage space in the case of high-speed networks. Therefore, trace reduction can be performed, which reduces the amount of data collected by applying packet filtering techniques. One may focus on exclusively capturing traffic belonging to a particular application which can be done using transport-layer port numbers. Alternatively, depending upon the technique used to classify traffic, one may only capture packets that request or establish a connection; or requires only the first few packets of a flow for analysis. Trace files can also be reduced: i) by storing the summary of a protocol-specific request of each application; ii) by capturing a limited amount of packets instead of complete flow packets; iii) by storing only the header information of TCP/IP protocol stack, or iv) by storing just the flow information instead of storing each packet information. Further, packet filtering can also be done using various packet sampling methods where packets are randomly (or pseudo-randomly) chosen for analysis purposes and should be chosen in such a way that they represent the traffic to a great extent which one wants to measure. The distinction of each sampling method depends upon the study purpose, state of the network, traffic characteristics, resource constraints, etc. Jurga and Hulb'oj in [25] and Duffield in [26] elaborated on the subject of packet sampling on traffic measurement.

1.4 Verification of ground truth of traffic

In the early days, traffic identification was an easy task that involved port-based identification by mapping transport-layer port numbers with the applications or signature-based identification by matching payload signatures with application protocols. But, as various Internet applications, especially P2P applications, evolved, the traditional approaches for traffic identification started becoming ineffective, as applications based on P2P architecture used random or well-known port numbers to hide their traffic. Hence, to address various issues involved in traffic identification, several new techniques based on statistical or behavioral methods have been developed and adopted over time.

It is essential to assess the ground truth application information of pre-collected traffic to test a new technique for traffic classification; otherwise, it has very limited value [27]. Due to privacy concerns, the packet traces available publicly only contain header information, making it challenging to verify the application's ground truth. But, this issue can be addressed if the packet traces are labeled for ground truth verification before headers are made available publicly. Another method that can be adopted is to manually verify the ground truth of traffic traces [28], but it is very slow and only feasible for smaller datasets. One may also assess the ground truth by using port number matching or payload inspection technique [29], but they have their limitations since port-based matching is inconsistent as many applications use random port numbers, whereas the DPI technique is ineffective if traffic is encrypted. Hence, using such approaches to find out the traffic's ground truth would produce inconsistent results while testing newer techniques. Due to such issues, researchers mostly collect their traffic traces to verify the ground truth of the applications and test the accuracy of their techniques; but such an approach gives inconsistent results while comparing various methodologies as their performance is evaluated under different conditions [30]. It is also possible to collect traffic traces from small computer networks which run pre-defined applications in a controlled environment, but the traffic properties generated by such an approach may not imitate human behavior. Some of the studies also tried to address the ground truth verification subject. Canini et al. [31] proposed a framework called GTVS to simplify and improve the application's ground truth, which uses the DPI mechanism and multiple heuristic rules. Gringoli et al. [32] presented a toolset called GT, which includes a deamon run on every client that returned the process information that initiated network connection. An identical approach based on clients was also proposed by Szabó et al. in [33].

None of the techniques proposed by various authors is perfect and have their own merits and demerits. Hence, the accuracy of the reference classification model will determine the performance of the new classification approach, which may lose its efficiency if there arise any change in the communication pattern of the applications. Therefore, a proper method should be chosen to assess the ground truth by looking at the capabilities and limitations of each, as this is one of the factors on which the quality of evaluation results depends.

1.5 Evaluation metrics for performance analysis

All network traffic classification techniques use some metrics to assess the classification results by correlating them with trace's ground truth information. Each case falls in one of the following categories:

a) *True Positive (TP):* It specifies that a case is correctly classified as associated with a specific class.
b) *True Negative (TN):* It specifies that a case is correctly classified as not associated with a specific class.
c) *False Positive (FP):* It specifies that a case is incorrectly classified as associated with a specific class.
d) *False Negative (FN):* It specifies that a case is incorrectly classified as not associated with a specific class.

A good classifier will minimize FP and FN. By using above mentioned metrics, various other metrics can be made for evaluating the performance of classifiers [34] [35], some of which may be equivalent, but many of them measure distinct classification aspects. Therefore, it is essential to know what is measured by a certain metric. The most commonly used metrics for traffic classification are defined in equation (*1.1*):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1.1)$$

Accuracy measures the capability of a classifier to identify positive and negative cases. It measures the overall efficiency of the classification model and hence shows its predictive power. But, relying only on accuracy to evaluate the classifier is insufficient if imbalanced datasets are used, which have many positive or negative cases, in which case the importance is given to the more popular class. Therefore, it is desirable to use some more metrics which can evaluate other aspects also. The most popular are Recall and Precision, which are used together for evaluating classifiers [36] and are defined in equations (*1.2*) and (*1.3*).

$$Recall = \frac{TP}{TP + FN} \qquad (1.2)$$

$$Precision = \frac{TP}{TP + FP} \qquad (1.3)$$

Recall measures the overall positive cases present in the dataset that are correctly classified by the classifier. It is also known as true-positive or hit rate. Precision measures the percentage

regarding the correctness of the positive cases that are identified by the classifier. It is also known as positive predictive value. Both precision and recall evaluate the ability to correctly identify positive cases by the classifier, but they also have a limitation. Both cases do not give information about the number of negative cases correctly classified by the classifier. Therefore, if required, then one can make use of another metric called Specificity [37] which can be used together with Recall for evaluation of positive and negative cases separately (in that case, Recall is usually called Sensitivity [38]) and is defined in equation *(1.4)*:

$$Specificity = \frac{TN}{FP + TN} \qquad (1.4)$$

Specificity measures the percentage of cases correctly identified by the classifier as negative. Karagiannis et al. [29] also defined another metric called Completeness, which they used together with Precision to refer to accuracy and is defined in equation (*1.5*):

$$Completeness = \frac{TP + FP}{TP + FN} \qquad (1.5)$$

Completeness determines the number of cases incorrectly or correctly classified as positive to the total positive cases in the dataset. Therefore, depending upon the objective of every classifier, proper metrics should be chosen to evaluate it. Table 1.1 presents the summary of various metrics along with their definition and the aspects they measure.

Table 1.1. Various evaluation metrics for performance measurement, where TP → true positive, TN → true negative, FP → false positive, FN → false negative.

Metrics	Defined as	Capability/Measures
Accuracy	(TP + TN) / (TP + TN + FP + FN)	Percentage of positive and negative cases correctly identified.
Recall	TP / (TP + FN)	Percentage of overall positive cases correctly identified
Precision	TP / (TP + FP)	Percentage regarding the correctness of positive cases identified
Specificity	TN / (FP + TN)	Percentage of negative cases correctly identified
Completeness	(TP + FP) / (TP + FN)	Percentage of positive cases correctly or incorrectly identified among overall positive cases.

1.6 Research Problem

More than a decade ago, peers used to communicate using client-server architecture on the internet, where clients (or peers) request data from the server and server responds, thus leading to asymmetric kind of traffic. With the proliferation of the internet, P2P applications/services

started emerging. Here every peer on the network acts as a client & server concurrently, thereby downloading the data from other peers and distributing the required data to the other peers at the same time. This leads to internet traffic going symmetric. With the rise in popularity of P2P applications/services as well as its number of users, P2P traffic has become one of the largest contributors of internet traffic, which have ended the dominance of various application protocols, namely: HTTP, FTP, SMTP, DNS, etc. that ruled the internet more than a decade ago [2].

As P2P traffic flows in both directions simultaneously, it produces a considerable amount of traffic in the network and hence consumes a lot of network bandwidth compared to client-server traffic. This poses an issue to the ISPs and network administrators as they need to supervise & control P2P traffic so that it alone does not consume available network bandwidth, thereby hampering the Quality of Service (QoS) of other network applications which use HTTP, HTTPS, FTP, SMTP, etc. protocols for communication. In addition to that, they also face challenges like conventional devices unable to handle large traffic, purchasing upstream bandwidth, costly backbone links, providing excellent broadband experience to customers, etc. From the network management perspective, ISPs or network administrators need to classify P2P traffic so that every network application gets its fair share of network bandwidth.

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) [36]. Nowadays, classifying P2P traffic with high accuracy is a difficult task since various P2P applications either masquerade or encrypt their traffic to avoid detection [18]. When the internet began, network traffic classification was an easy task that could be achieved using a simple and easy-to-implement approach called the port-based technique. This technique could easily classify legacy applications like HTTP, FTP, DNS, etc., with high accuracy, but it soon became inefficient/inaccurate in traffic classification since P2P applications started using random port numbers for communication to navigate through firewall & other network restrictions. Therefore, another approach called the payload-based technique was adopted, which classifies network traffic by inspecting its packet payload. Such an approach has high accuracy in classifying network traffic, but despite this fact, it cannot be applied in various situations such as unavailability of payload information, traffic payload encryption, etc.

To overcome various limitations of traditional classification techniques, nowadays modern classification technique called Classification in the Dark is adopted, which classifies

P2P traffic either by using statistical features of traffic flows or by observing behavioral patterns associated with the traffic flow [18]. The accuracy of this technique relies heavily on the robustness of statistical features or behavioral patterns selected for classification.

1.7 Research Objectives

The primary purpose of this thesis is to propose a classification model to achieve the following objectives:

a) Analyze the behavior of Peer-to-Peer network traffic to improve its identification accuracy.
b) Uniquely identify traffic of web-service incorporating file-downloads from Peer-to-Peer network traffic and improve its identification accuracy by analyzing and varying the traffic features.
c) Uniquely identify traffic of web-service incorporating video-streaming from the Peer-to-Peer network traffic and improve its identification accuracy by analyzing and varying the traffic features.

1.8 Benefits of P2P traffic classification

Classification of P2P traffic servers various purposes for ISPs and network administrators; some of which are mentioned below:

- Network-specific policies can be implemented for providing QoS to every network applications.
- Billing mechanism can be implemented based on the type of traffic generated by customers.
- Network security measures can be implemented.
- Issues of network congestion can be addressed.
- Network planning for public or campus area network can be performed.
- Such traffic can be prioritized, limited, or completely banned for maintaining QoS or evade network congestion.

1.9 Suggested Approach

Due to various limitations of traditional traffic classification techniques, nowadays Classification in the Dark approach is employed, as it is effective in classifying P2P traffic. It can classify encrypted traffic and unknown applications from target classes also, but it cannot

perform traffic classification with as high accuracy as the payload-based technique [18]. In addition to that, 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, which employs Classification in the Dark approach. The proposed technique utilizes the benefits of both heuristic-based and statistical-based techniques to classify P2P traffic with high accuracy.

1.10 Thesis Organization

The remaining thesis chapters have been organized as follows:

Chapter 2 introduces various network traffic classification techniques and literature survey, which discusses the related works proposed by various researchers for classifying P2P network traffic.

Chapter 3 discusses a multi-level P2P traffic classification approach that classifies network traffic either as P2P or non-P2P. The proposed approach uses the combination of heuristic-based & statistical-based techniques to classify the network traffic either as P2P or non-P2P.

Chapter 4 discusses a 2-step traffic classification approach that specifically classifies P2P file-sharing traffic (i.e., P2P traffic involving file downloading/uploading) from P2P traffic as a whole, which is generally observed when P2P peers share files/data directly over the internet. The proposed approach employs the combination of heuristic-based & statistical-based techniques to classify P2P file-sharing traffic uniquely.

Chapter 5 discusses a 2-step traffic classification approach that specifically classifies P2P-VoIP (video) traffic (i.e., P2P traffic involving video-streaming) from P2P traffic as a whole, which is generally observed when P2P peers are involved in communication during video conferencing or online meetings. The proposed approach employs the combination of heuristic-based & statistical-based techniques to classify P2P-VoIP (video) traffic uniquely.

Chapter 6 presents the conclusions and future work recommendations.

CHAPTER 2

BACKGROUND KNOWLEDGE

2.1 Introduction

This chapter provides information about various techniques to classify internet traffic along with their advantages & limitations. First, the background knowledge about traditional traffic classification approaches is presented. Then, modern traffic classification approaches adopted nowadays by the researchers, along with various studies, are discussed.

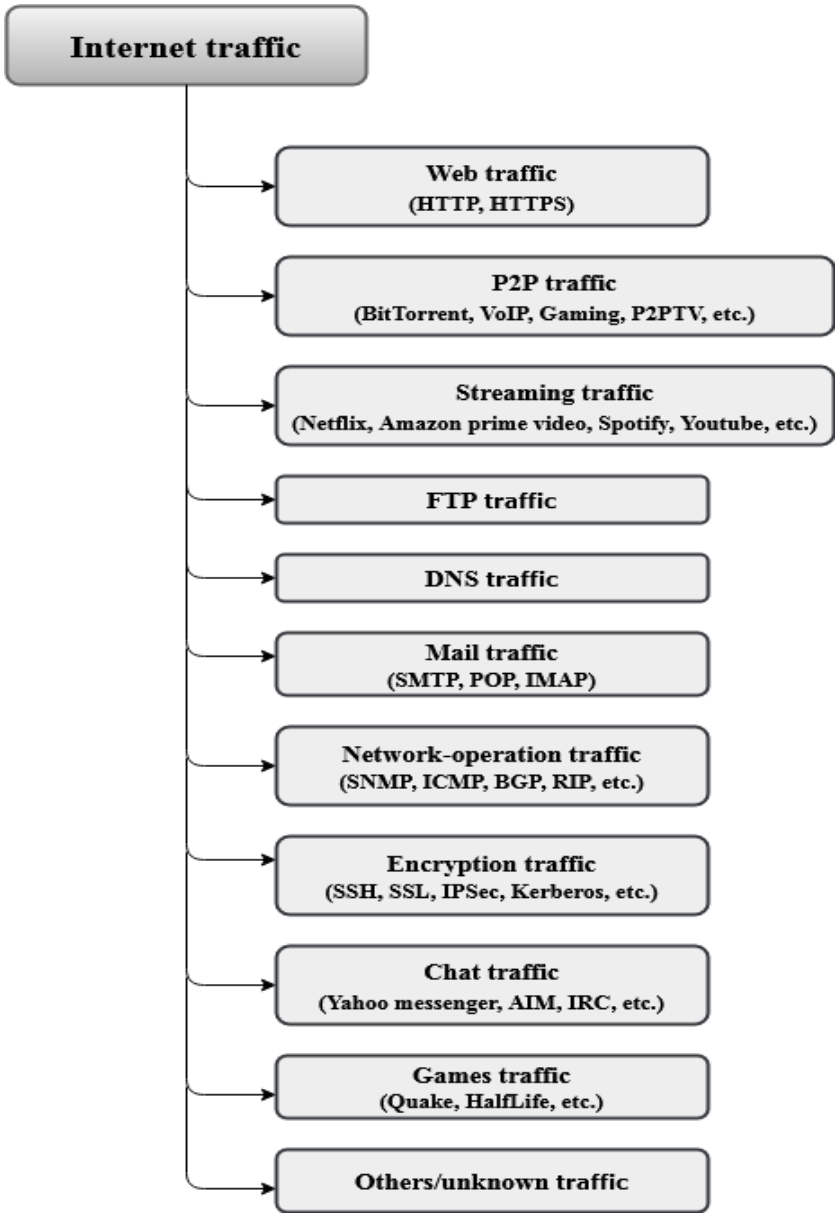


Figure 2.1. Various categories of internet traffic.

Various kinds of applications generate traffic over the internet. Hence, it can be seen that internet traffic consists of multiple categories as shown in Figure 2.1, such as Web traffic (e.g., HTTP, HTTPS), P2P traffic (e.g., BitTorrent, VoIP, Gaming, P2PTV, etc.), Streaming traffic (e.g., Netflix, Amazon prime video, Spotify, Youtube, etc.), FTP traffic, DNS traffic, Mail traffic (e.g., SMTP, POP, IMAP), Network-operation traffic (e.g., SNMP, ICMP, BGP, RIP, etc.), Encryption traffic (e.g., SSH, SSL, IPSec, Kerberos, etc.), Chat traffic (e.g., yahoo messenger, AIM, IRC, etc.), Games traffic (e.g., Quake, HalfLife, etc.) and other/unknown traffic.

The prime objective of this research work is to focus on P2P traffic classification. We have chosen P2P traffic for classification since P2P applications have become very popular since the past decade. Various P2P applications either masquerade or encrypt their traffic to avoid detection and the traffic generated by such applications continues to grow as new applications keep emerging and many peers join the network to use them. This leads to network congestion. Therefore, such kind of network traffic needs to be monitored and controlled so that P2P traffic alone doesn't consume a large portion of the available bandwidth.

Broadly, internet traffic can be categorized into two groups: P2P & non-P2P traffic, as shown in Figure 2.2. P2P traffic further consists of multiple categories such as File-sharing traffic (e.g., BitTorrent, eMule, etc.), Voice over Internet Protocol (VoIP) traffic (e.g., Skype, Google-meet, etc.), and other P2P traffic such as P2PTV, Gaming, etc. P2P file-sharing traffic involves sharing & distribution of data using P2P architecture, where users can share & distribute digital data such as books, media, documents, software, games, etc., among other peers on the network. P2P VoIP traffic involves the transfer of voice & video data (with better quality), where users can directly connect & communicate with each other. There exist other kinds of P2P traffic as well; for example, P2PTV traffic (e.g., PPLive, QQLive, etc.), which uses P2P architecture to redistribute video streams which are typically TV channels in real-time; P2P gaming traffic (e.g., World of Warcraft, For Honor, etc.) which uses P2P architecture to connect peers directly for performing online gaming, etc.

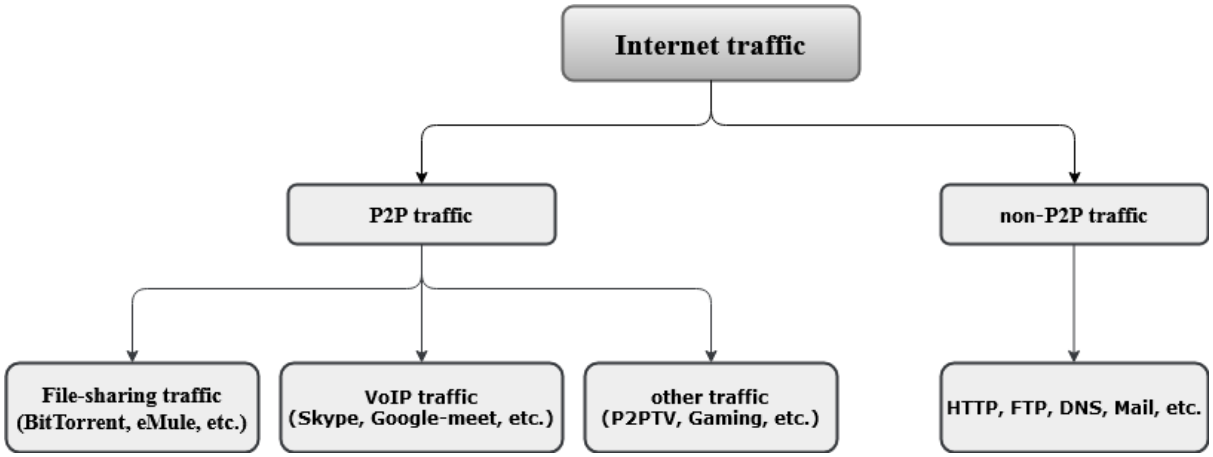


Figure 2.2. Internet traffic categorization as P2P & non-P2P.

Traffic generated by non-P2P applications typically possesses client-server behavior where the client initiates TCP connection to the server for requesting the data, and the server responds to this request. A general web-browsing session consists of a 3-way handshake process, where the TCP "SYN" flag is initiated by the client to port 80 or 443 of a web server, and then the server acknowledges it with TCP "SYN-ACK" flag. Thereafter, the client sends an acknowledgment "ACK" flag back to the server. After the completion of this process, the client requests the desired data from the server. It can be noticed that the host which requests the data is always the client and the host which provides the response is always the server. Many web applications communicate in this manner. There exist some applications which have client-server behavior and communicate with port numbers other than 80 or 443. For example, the File transfer protocol (FTP) utilizes client-server architecture for communication where one host sends the data, and the other host receives it. Two channels are used by FTP for communication, where one channel is utilized for sending control signals over port number 21, and the other is utilized for transferring the data over port number 20. A cloud-based application named Dropbox also possesses client-server behavior. A Dropbox client uses HTTPS protocol to request files from a server and then download them on a local machine. It is different from FTP protocol in the manner that it can sync any shared files using port number 17500 in the Local Area Network (LAN) without contacting the server; since it will first look for new files on the LAN; thereby bypassing the need to connect to Dropbox server to download the file.

P2P applications behave differently in comparison to client-server applications. The main motivation behind P2P architecture is to maximize file-sharing benefits amongst collaborating peers. The hosts running P2P applications act as a client as well as server concurrently; hence transfer & receive the data at the same time. Therefore, traffic from P2P applications can be

observed traveling in both directions in huge amounts as all peers upload & download the data simultaneously. This makes the network traffic symmetric which consumes a lot of network bandwidth. Client-server applications, in contrast, are asymmetric where upstream traffic is very less in comparison to downstream traffic.

Earlier, traffic identification and hence classification was an easy task where traffic could be easily identified using the port-based technique. However, as the P2P architecture (and hence its applications) evolved, it started using random port numbers or the port numbers assigned to various well-known protocols (such as HTTP), resulting in inaccurate traffic classification. Due to this fact, another technique based on inspection of payload known as Deep Packet Inspection (DPI) was adopted. It is the most accurate technique to classify network traffic, but it also possesses various limitations. So, nowadays, researchers use modern classification technique (known as Classification in the Dark), which focuses on the statistical-based or behavior-based approach to classify network traffic; and hence overcomes various limitations which are present in traditional classification techniques. The following sections elaborate on different types of traffic classification techniques along with their merits and demerits.

The remaining chapter elaborates on traditional as well as modern traffic classification techniques along with their merits and demerits and is organized as follows. Section 2.2 discusses traditional traffic classification techniques, namely port-based traffic classification. Section 2.3 discusses another traditional traffic classification technique, namely payload-based traffic classification. Finally, Section 2.4 discusses modern traffic classification techniques, namely Classification in the Dark.

2.2 Port-based traffic classification

This technique relies on the identification of application protocols which use TCP or UDP port numbers since each application is associated with well-defined port numbers, which are defined by Internet Assigned Numbers Authority (IANA) [39]. For example, HTTP traffic utilizes port number 80, DNS traffic utilizes port number 53, and SMTP utilizes port number 25. This is a simple technique as it relies on packet headers only to extract port numbers from it. A classifier placed in the middle of the network analyses for the SYN packets (which are TCP packets used for a 3-way handshake to establish a connection) to know about the server-side of a TCP connection and hence identifies the type of traffic flowing through the network by looking at TCP SYN packet's target port number in IANA's registered list of port numbers [39].

Similarly, UDP traffic can be identified using the port numbers it uses during communication between the hosts, but here connection establishment or its maintenance does not take place.

Table 2.1. Various P2P protocols using well-known port numbers.

Protocols	TCP Ports	UDP Ports
AIM - messages	5190	5190
AIM - video	1024–5000	1024–5000
ARES Galaxy	32285	32285
BitTorrent	6881–6999	
Blubster	41170–41350	41170–41350
Direct Connect	411, 412, 1025–32000	1025–32000
eDonkey	2323, 3306, 4242, 4500, 4501, 4661–4674, 4677, 4678, 4711, 4712, 7778	4665, 4672
FastTrack	1214, 1215, 1331, 1337, 1683, 4329	
Gnutella	6346, 6347	6346, 6347
GoBoogy	5335	5335
HotLine	5500–5503	
ICQ	5190	
iMesh	80, 443, 1863, 4329	
IRC	6665–6669	
Kazaa	1214	1214
MP2P	10240–20480, 22321, 41170	41170
MSN	1863	
MSN - file transfer	6891–6900	
MSN – voice	6901	6901
Napster	5555, 6666, 6677, 6688, 6699–6701, 6257	
PeerEnabler	3531	3531
Qnext	5235–5237	5235–5237
ROMnet	6574	
Scour Exchange	8311	
ShareShare	6399	6388, 6733, 6777
Soribada	7675–7677, 22322	7674, 22321
SoulSeek	2234, 5534	2234, 5534
WASTE	1337	1337
WinMX	6699	6257
XMPP / Jabber	5222, 5269	5222, 5269
Yahoo – messages	5050	
Yahoo – video	5100	
Yahoo – Voice	5000–5001	5000–5010

The main advantage of this technique is that it doesn't involve any calculations and hence is fast to identify network traffic. Also, its implementation is simple, which requires the addition of port numbers in the database for new applications that have recently emerged. However, with the evolution of the internet, this approach started to become obsolete [8] [40] [41] as some applications such as P2P started using dynamic port numbers and port numbers that may not be registered with IANA (e.g., Napster and Kazaa) [42]. Gomes et al. [18] presented a list of TCP & UDP port numbers utilized by several well-known P2P protocols, which is shown in Table 2.1. Further, to get through the firewall, many applications masquerade by hiding their traffic behind well-known port numbers such as port number 80, which maps to HTTP traffic. This technique fails if there is encryption at the IP layer, which obfuscates TCP or UDP port numbers, hence making it impossible to recognize actual port numbers utilized by the applications. Earlier, some P2P applications utilized port numbers or ranges which were used to identify P2P application protocols. Moore and Papagiannaki [40] identified that byte-accuracy of at most 70% could be achieved using the port-based classification technique. Madhukar and Williamson [8] showed that the port-based technique could not classify internet traffic correctly. Karagiannis et al. [7] 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.

2.3 Payload-based traffic classification

This technique is usually most accurate and is based on inspecting packet headers and packet payloads. It relies on a database that contains signatures of previously-stored application protocols. The packet payload is inspected bit-wise to locate the bit-stream that contains the signatures (which are predefined byte sequences) of an application protocol. Hence, the traffic can be identified accurately when packet signatures of network applications match with stored signatures in the database. For example, the 'xe3\x38' string is contained in eDonkey P2P traffic, the '\GET' string is contained in web traffic, and so on. This technique is not only employed for P2P traffic identification [41] [43] [44] but also in scenarios that involve the identification of threats such as network intrusion detection [45], malicious data, and other traffic anomalies. Such technique is also significant for accounting solutions and charging mechanisms, where accuracy is crucial.

Song and Zhou [46] proposed a file-aware P2P traffic classification mechanism based on the DPI technique to identify a file and flows associated with it, which consists of two strategies

based on i) per-file bandwidth consumption and ii) the number of per-file concurrent active flows. This approach maintained 6-tuple (source-IP, source-port, destination-IP, destination port, protocol, and file-id) file-level information in the flow table. To reduce the computational overhead involved in the traditional DPI technique, pattern matching (involving only simple pattern-sets) occurred at the beginning of the payload, and depth of inspection involved only a dozen of bytes. Authors evaluated their approach on a dataset collected from the campus network, where the majority of P2P applications include: BitTorrent, eDonkey, and Gnutella, and their ground truth was verified using GTVS. The proposed approach achieved 100% accuracy and completeness, ranging from 88-93%.

The prime advantage of the payload-based technique is that it performs network traffic identification fairly accurately. However, it also suffers from various limitations. It involves a significant amount of complexity and processing load on network equipment which is used to identify network traffic. Such a technique is unfeasible in high-speed networks. Hence to resolve this issue, some mechanisms inspect small number of packets of each flow only, which is a trade-off between accuracy & efficiency, and sometimes in such cases, signatures may not be contained in the part that is captured, which may lead to inaccurate identification of traffic. The database or the device needs to be kept updated with signatures of newly emerged application protocols, or else some new traffic may get unidentified. Furthermore, it is difficult to maintain signatures with a high hit and low false-positive ratio. For example, payloads of both Gnutella and HTTP traffic contain the '\GET' string and hence raise ambiguity. The major drawback of this technique is that identification of network traffic becomes almost impossible if traffic is encrypted or if traffic contains proprietary protocols. Direct analysis of packet payload may also breach the privacy policies of some organizations or violate relevant privacy legislation.

2.4 Classification of traffic in the Dark

As various limitations exist in the port-based and payload-based techniques, therefore new approaches have been developed and adopted which do not rely on port number and inspection of payload to identify the traffic. Such an approach is often called Classification in the Dark [29] [47], which classifies the traffic using generic properties of packets [28] such as packet size, total bytes sent, ports, etc., or by observing behavioral or statistical patterns of the flows. The prime advantage of this technique is that it can classify the traffic without inspecting payload or relying on port numbers. However, it is not as accurate as the payload-based

technique, but recent studies have achieved good accuracy in classifying the traffic. Also, this approach applies to any unknown application since methods based on it classify the traffic in a particular class instead of identifying specific applications. Various methods which fall under this approach are discussed as follows.

a) **Statistical or behavioral signatures:** Such method rely on packet or flow level properties of traffic such as packet size, totals bytes sent or received, flow duration, flow size, packet inter-arrival time, TCP or UDP ports used, etc.; which can be used individually or collectively for calculation of statistical properties such as variance, average, and probability density function. Such a method needs a preliminary learning phase for building a reference model to classify the traffic.

Freire et al. in [48] and [49] proposed a technique to identify VoIP traffic concealed in Web traffic by analyzing various network-data properties, which are: the size of Web request and response, number of per-page requests, inter-arrival time between requests, and retrieval time of page. They evaluated their approach on VoIP data of Google-Talk and Skype, which was collected from ISP and university links, and achieved recall rates of about 90% for VOIP calls and 100% for VoIP calls concealed in Web traffic.

Gomes et al. [50] analyzed the behavior patterns of several P2P & non-P2P applications and found that there is high heterogeneity in P2P packet sizes in comparison to non-P2P traffic. Heterogeneity degree was represented using entropy, and its value was calculated using a sliding window that contained a fixed number of packets. It was found that P2P traffic related to VoIP applications had high entropy values, whereas regular client-server traffic had consistently smaller entropy values.

Sun and Chen [51] proposed a novel technique suitable based on the C4.5 decision tree for identifying the application associated with a TCP flow, using two characteristics: the ACK-Len ab and ACK-Len ba, which are the data volume first sent by communicating parties continuously. Using this approach, authors classified four different types of applications: WWW, FTP, Email, and P2P; where P2P traffic was identified by analyzing that both parties involved in communication send considerable volumes of data to each other, thus reflecting P2P behavior. Three datasets were used, where first was taken from Moore [52], second from the working environment (called Set1), and third was extracted from Set1 by using characteristic mentioned in ref. [53]. The proposed approach can be used for online traffic classification as it only depends on data's total length of the first

few packets on the flow, which greatly save storage space and classified P2P traffic with accuracy, recall, and precision rates ranging from 97.648 to 99.694%, 30 to 80% and 65 to 93%, respectively.

He et al. [54] proposed fine-grained host-based P2P traffic classification by simply counting particular flows (i.e., clustering flows). This approach locates all P2P hosts within the monitored network and identifies the types of P2P applications running. It builds application profiles of each P2P application by using the flow information that describes its most significant network activity pattern and is learned from traffic traces generated by the corresponding P2P application. The performance is evaluated on traffic datasets consisting of P2P applications such as BitComet, BitTorrent, eMule, Vagaa, and Thunder. Verification of ground truth was done manually by investigating each host running a P2P application. The experimental results achieved average true-positive & false-positive rates of 97.22 & 2.78%, respectively. The proposed approach did not use complicated statistical features of traffic or machine learning algorithms and can easily accommodate new P2P applications in scope of classification. It is also able to classify encrypted traffic in real-time.

Yang et al. [55] proposed a method to identify P2P live streaming based on union features by analyzing its behavioral characteristics. The datasets consisted of a mixture of traffic from BitTorrent and Thunder, which are file-sharing applications, and traffic from PPTV, PPStream, QQlive, and UUSE, which are on-demand and live streaming applications. The experimental results achieved 95 % accuracy in identifying P2P live streaming traffic.

Qin et al. [56] developed a framework named CUFTI for identifying and managing core users' P2P traffic (i.e., long-lived peers). They studied peer's lifetime in the PPlive system and identified core users from the overlay. The model utilized payload length and direction of the first few control packets of different P2P applications (PPlive, BitTorrent, and Thunder) as statistical features that were extracted using the longest common subsequence (LCS) and performed flow identification. The experimental results achieved false positive and false negative rates of 3.49 and 8.47 %, respectively, in identifying PPlive traffic. Further, the model can be utilized for real-time identification of traffic.

Zhang et al. [57] proposed a component-based method to detect P2P traffic using UDP for communication. In graph theory, a component is defined as connected sub-graphs from

a disjoint graph. The approach uses graph-level statistics to detect P2P traffic (utilizing UDP) and does not use packet-level information. The dataset consisted of records taken from Netflow version 5 and exported from university campus network border-link.

b) **Heuristic-based methods:** This method classifies the traffic by observing the behavioral patterns of traffic using a pre-defined set of heuristics such as hosts behaving as a client & server simultaneously, number of connections made by the host, number of different addresses or ports a host connects to, hosts using both TCP & UDP for communication, etc. A set of heuristics is analyzed sequentially, and the packets or flows are classified as associated with a particular class depending upon the obtained results. There exist some studies that utilize heuristics to identify P2P traffic.

Per'enyi et al. [58] proposed a technique for identification of P2P traffic that utilizes a set of six heuristics: usage of UDP and TCP simultaneously, well-known P2P ports, number of consecutive connections existing between two peers, several flows having the same flow identities, flow duration greater than 10 min or flow-size greater than 1 MB, and an IP address using the same port number for more than five number of times during analysis. A small labeled traffic trace was used for validation of this approach, which achieved a recall rate of 99.14% for P2P traffic and 97.19% for non-P2P traffic.

John and Tafvelin [59] redefined the combination of heuristics used in [58] and [44] and proposed the heuristics: usage of UDP and TCP simultaneously; well-known port numbers of P2P protocols; the port numbers that are generally used; the relationship between the number of ports and IP addresses; flow-duration greater than 10 min or flow-size greater than 1 MB. They collected the traffic traces from university links and achieved a recall rate of 98%.

Hong [60] proposed a novel method to identify P2P traffic using UDP protocol and revealed & validated three unique characteristics that will not appear together in TCP or UDP traffic produced by non- P2P applications, which are: i) nearly all UDP traffic of a local peer transfers data using a fixed port number; ii) nearly all distant peers use a single port number for communication with a local peer, and iii) size of UDP packets produced by P2P applications is relatively fixed. These characteristics were examined by collecting 100 blocks of P2P traffic (consisting of BitSpirit, Emule, and other P2P applications), each ranging from 100 M bytes to 200Mbytes, and evaluation of this approach achieved an accuracy ranging from 98.4 to 99.6%.

Reddy and Hota [61] proposed a new set of heuristics to identify P2P hosts based on their connection patterns, and they do not require any payload signatures. The datasets used were realistic in nature ad consisted of applications, namely HTTP, FTP, Dropbox, SMTP, eMule, Frostwire, Skype, uTorrent, and Vuze. The authors verified their approach in real-time, and only 0.2% of P2P traffic remained unclassified. As their approach consisted of minimal heuristics, it can be used for real-time identification; but it can only identify coarse-grained P2P applications instead of fine-grained P2P applications.

Bashir et al. [62] proposed an approach based on heuristics to identify BitTorrent activities using Netflow records by observing three major segments of traffic: a) traffic from peers contacted via DHT, b) TCP traffic from peers contacted via trackers, and c) UDP traffic from peers contacted via trackers. The approach was tested on five real-life datasets having a mixture of applications consisting of BitTorrent, P2P radio streaming application, Skype, SopCast, and PPStream. The experimental results achieved the byte accuracy ranging from 91.3 to 95.4% in identifying BitTorrent activity.

The heuristic-based technique is application-specific since classification heuristics & models need to be built for every application which is to be classified. The main advantage of this technique is that it has comparatively higher accuracy than other modern classification techniques since it performs classification by exploiting traffic features that are unique to each application. Therefore, it is most suitable in classifying applications that share common flow features and having unique connection patterns/behavior. However, the drawback of this technique is that the classification models built using this approach lose accuracy if applications evolve or change their communication patterns/behavior over time.

c) **Machine Learning methods:** In the machine learning approach, a ML algorithm is employed, which performs traffic classification by identifying the key features that distinguish various types of traffic from each other. Machine learning algorithms make use of multiple traffic features (which acts as signatures) and compare them with the feature-set of trained data which is already labeled with a particular class, for identifying various classes of traffic. This process helps in classification by determining the likelihood of testing data belonging to a particular class of traffic. The machine learning approach consists of two categories: supervised machine learning & unsupervised machine learning. In supervised learning, during the training phase, a classification model is built using the training dataset. The ML algorithm generates a classifier model by analyzing the relationship between the traffic flow features and the output class value, which then

predicts the type of traffic flow by analyzing its statistical features. In the testing phase, statistical features of a traffic flow are extracted and fed into the classifier model. If the characteristics of a flow match the distinct characteristics of a particular class, then it is classified accordingly. Traffic flow features are the numeric values that are calculated over numerous packets belonging to that particular flow. In unsupervised learning, the ML algorithm produces a reference model by taking the un-labeled traffic dataset and derives the correlation between various dataset items. This model is then utilized to classify the testing data (i.e., unknown traffic). Machine learning techniques based on supervised or unsupervised methods have been adopted in various studies such as clustering [63], Bayesian estimators or networks [64], and decision trees [65]; which work on a set of traffic characteristics by correlating them using probability functions and hence classify the packets or flows as belonging to a particular class. Some of the studies are mentioned below:

Mohammadi et al. [2] proposed a hybrid classification approach using a genetic algorithm to classify P2P traffic. The genetic algorithm was used in calculating the minimum classification error (MCE) matrix, which is then used to map dataset features into a new space where they can be easily be classified into separate classes. The mapped dataset is fed into a classifier named neural networks. Three different indexes, namely mutual information, Dunn, and SD, were measured to compare the proposed methodology with standard MCE-based & regular (i.e., with any feature mapping) approaches. The experimental results showed that the proposed approach reduces overlap among classes and gives improved classification accuracy of 96%.

Schmidt and Soysal [66] proposed a technique involving Bayesian network to identify P2P traffic by using the parameters: well-known port numbers, IP packets per-flow distribution, packet-size distribution, octets-per-flow distribution, and flow-time distribution. They collected the traffic from the academic network to evaluate the performance of the classifier in their technique as well as in signature-based technique and showcased the results of false-positive ranging between 22 to 28% and false-negative ranging between 16 to 26%.

Cao et al. [67] proposed a technique using Classification And Regression Tree (CART) for real-time identification of application protocols at both flow-level and host-level. They collected the traffic traces of HTTP, SMTP & FTP from enterprise networks by port number filtering method, and traces of BitTorrent were collected actively at the home

environment in a controlled manner to assess the ground truth. By evaluating this technique, the classification results obtained showed false-positive rates ranging from 0.05 to 12.7% and false-negative rates ranging from 0 to 17.9%.

Raahemi et al. [38] proposed a technique using a set of network-level packet attributes to identify P2P traffic by using Concept-adapting Very Fast Decision Tree (CVFDT). To evaluate the performance of their technique, they used labeled datasets and achieved accuracy ranging from 79.50 to 98.65% and specificity ranging from 82.96 to 95.89%.

Angevine and Zincir-Heywood [68] classified TCP and UDP flows of Skype using C4.5 decision tree and AdaBoost algorithms. They collected the labeled traffic traces from the university network and achieved a recall rate ranging from 94 to 99 % with their technique.

Wang et al. [69] identified traffic of multiple P2P protocols using a classifier based on a decision tree called Random Forest. They captured the traffic traces from academic and residential networks and evaluated their technique using a manually labeled dataset to achieve accuracy ranging from 89.38 to 99.98% and precision ranging from 32.69 to 100%.

Dainotti et al. [70] presented a classification technique based on hidden Markov models and using parameters: packet size & inter-packet time. They carried out classification on real-traffic traces of MSN messenger, eDonkey, HTTP, SMTP, P2P-TV, PPlive & two multi-player games, whose traces were verified manually as well as using DPI technique, to achieve recall rates ranging from 90.23 to 100%.

Valenti et al. [71] adopted a mechanism based on Support Vector Machine (SVM) and the number of packets sent back & forth between the peers during a short interval of time; to identify P2P-TVapplications. They tested their approach on traffic captured in larger test-bed to achieve recall rates ranging from 91.3 to 99.6 %.

Liu et al. [72] proposed a mechanism by utilizing a supervised ML algorithm and ratio of the amount of downloaded & uploaded traffic in every minute as a recognition pattern. They classified P2P applications of Maze, PPlive, BitTorrent, eDonkey, and thunder and achieved accuracy ranging from 78.5 to 99.8%.

Raahemi et al. [73] identified P2P traffic using the neural network: Fuzzy Predictive Adaptive Resonance Theory, which was built by utilizing IP headers data. This approach utilized labeled datasets to achieve classification accuracy ranging from 78 to 92%.

Hu et al. in [74] [75] presented a novel approach to identify the various applications by building behavioral profiles using association rule mining. They extracted flow statistics by selecting five flow tuples and correlated them using the Apriori algorithm. The authors collected the traffic traces from an on-campus network, which were verified manually as well as using the DPI technique and tested this mechanism on BitTorrent and PPlive to achieve the recall rates ranging from 90 to 98%.

Liu and Sun [76] proposed a new approach called P2PTIAL that doesn't require a fully labeled samples-set for P2P traffic identification by active learning, which consists of two parts: Support Vector Machine (SVM) and uncertainty selection policy. SVM acts as a learner, which repeats the learning process on both labeled & unlabelled samples, whereas uncertainty selection (which is based on distance) selects unlabelled samples to be labeled by an oracle (e.g., a human annotator). Further, to improve its effectiveness, the authors employed the Support Vector Data Description (SVDD) technique to filter unlabelled samples having little contribution in active learning to reduce storage space & save computation cost; and used unlabeled samples pre-labeled information to avoid imbalanced learning. They utilized Moore-dataset [28] [64], which includes traffic from applications: P2P, WWW, Bulk, Database, Interactive, Mail, Services, Attack, Games & Multimedia and evaluated their technique on both un-balanced & balanced learning to achieve the accuracy rate ranging from 79.65 to 86.86 % and 93.00 to 93.07 %, respectively.

Jiang and Tao [77] proposed a P2P traffic identification model based on SVM that can work on encrypted traffic and selected three characteristics: i) change of the mean square value of packet size, ii) average flow duration, and iii) ratio of IP address and port numbers. The performance achieved in terms of precision, false-positive and false-negative rates range from 96.55 to 97.89%, 2 to 2.8%, and 2.45 to 5.29%, respectively.

Gong et al. [78] proposed an improved SVM incremental learning algorithm for P2P traffic identification, which can save storage space and increase identification accuracy (87.89%) when its performance is compared with standard SVM incremental learning algorithm (having 80.35 % accuracy) and SVM-based retraining algorithm (having 78.90 % accuracy) for an increased number of test samples.

Deng et al. [79] proposed the ensemble learning model, which integrates Random Forests and feature-weighted Naive Bayes for P2P traffic identification. Network traces

considered for evaluation consisted of both P2P traffic (BaiDuYingYin, BaoFengYingYin, PPS, PPlive, QQlive, XunLeiKanKan, and Thunder) and non-P2P traffic (Web, Youku, and Souhu) and achieved an accuracy of 92.47%; which overall performs better when compared to simple machine learning methods.

Jie et al. [80] proposed a novel and fine-grained P2P traffic classification approach that relied on the count of most frequent and steady flows generated by corresponding P2P applications called Clustering Flows. This approach exploited only basic properties of flows (protocol, packets size, and number) to perform the classification using the SVM algorithm and doesn't require any other complicated traffic statistical or behavioral features. The experiment performed on traffic traces of P2P applications include BitTorrent, eMule, PPTV & Cbox and, achieved a true-positive rate ranging from 95.4 to 98.63% and a false-positive rate of 0.01%.

Bozdogan et al. [81] evaluated the performance of machine learning algorithms for the classification of P2P applications, which include BitComet, uTorrent, and BitTorrent. Four supervised algorithms (C4.5, Ripper, SVM, and Naïve Bayes) and one un-supervised algorithm (K-means) were evaluated using the metrics: detection rate, false-positive rate, f-measure, and correct classification rate. The experimental results showed that the Ripper algorithm performs better in identifying P2P network traffic.

The major limitation of ML methods is that the accuracy of classification results completely relies on the accuracy & quality of given training datasets upon which the machine learning algorithms rely to perform traffic classification. Also, it is very difficult to obtain a flawless training dataset that can be used for classifying various kinds of traffic (irrespective of their origin) because different networks operate differently. Another limitation of the machine learning technique is that it becomes almost difficult to accurately classify two or more traffic classes if they share similar characteristic features. This is because if there are not enough distinct features capable of distinguishing various similar classes, then in such a case, the traffic features of such classes will overlap with each other leading to the machine learning algorithm getting biased towards one of the traffic classes, which leads to inaccurate results.

2.4.1 Classification of traffic using combined approaches

There also exist some studies which combine different classification approaches to identify network traffic, which are discussed below. The basic purpose of combining various

approaches for classifying the traffic is to utilize the benefits of different classification techniques.

Karagiannis et al. [44] adopted a cross-validation mechanism to identify traffic from FastTrack, eDonkey, Gnutella, BitTorrent, Direct-Connect, MP2P & Ares; by using port-based, payload-based, and behavioral patterns techniques. In addition to using payload signatures for particular applications, two heuristics were used to identify flows belonging to P2P applications. They are (i) identifying pairs of source & destination IP addresses using both TCP & UDP protocols; and (ii) identifying the number of unique IP addresses that are connected with the destination-IP with an equal number of unique ports of the destination. The behavioral approach achieved recall rates ranging from 90 to 99%. Also, they compared the performance of the payload-based approach with the behavioral-based approach and found the false-positive rates ranging from 8 to 12% of overall P2P traffic.

Dedinski et al. [82] adopted an approach of P2P traffic identification that utilizes active crawlers for collecting information of peers of a specific application to infer the overlay network topology. Besides, while analyzing behavioral patterns, the authors used the wavelet analysis technique on traffic to analyze network-level properties: per-packet or inter-packet arrival times. The performance of this architecture evaluated on traffic belonging to eDonkey and FTP.

Adami et al. [83] proposed a real-time mechanism using payload-based method & statistical method to identify different Skype clients in the network, which have the communication of file transfer, direct calls, calls to phone service, and calls using relay nodes. They collected the traffic traces from a university network and ADSL link of a small network. The performance of this mechanism (which was conducted both online & offline) was tested for both TCP & UDP with the other five classifiers. It achieved false-positive rates ranging between 0 to 0.01% and false-negative ranging between 0.06 to 0.64%, in terms of bytes & flows.

Yan et al. [84] proposed a novel technique for P2P identification based on host heuristics & flow statistics. To find out if a host is participating in P2P application, authors first matched its behavior with pre-defined heuristic rules:- IP popularity ratio, port-pair difference, ephemeral-port ratio, failed-connection ratio; and secondly refined the identification by comparing each flow's statistical features:- flow-bytes & flow-duration, and byte-ratio of forward & backward direction. The traffic traces were gathered at the campus network's edge

router. They consisted of the traffic traces of HTTPS, HTTP, POP3S, POP3, IMAPS, IMAP, BitTorrent, Skype, and eDonkey. This proposed technique achieved the flow & byte accuracy of 93.9% and 96.3%, respectively.

Ye and Cho [85] presented a 2-step hybrid classifier that combines packet-level and flow-level classifiers to classify P2P traffic. The first step (which is packet-level classification) is the combination of signature-based and heuristic-based techniques, where the packets, if not classified with the former approach, are checked with the latter one for classification. The second step (which is flow-level classification) is based on the combination of statistical & pattern-heuristics approach; which is applied to the traffic that remains unclassified in the first step. The authors used the REPTree algorithm with a statistical approach after comparing six ML algorithms for their performance. They then applied pattern heuristics to improve the results achieved by the former approach. Four datasets were used for evaluation of this technique; where the first two were taken from the University of Brescia and Ericsson Research in Hungary other two in a controlled environment inside the Dankook University that was labeled with actual application types. The proposed scheme showed a slight overhead with high scalability and achieved accuracy rates of 98.19 & 99.82% in terms of flows & bytes, respectively.

The authors in [86] used a similar hybrid approach to classify and distinguish between P2P botnet traffic from P2P traffic. The botnet traffic of Storm, Waledac, Conficker, C&C, and Zeus was mixed to create three datasets. The proposed approach provides low overhead and achieved flow & byte accuracy of 97.10% & 97.06%, respectively, using real datasets.

Wang et al. [87] proposed a novel Application Behavior Characterization technique for P2P identification. It extracts behavioral features (number of external IP addresses, number of flows, number of bytes, and number of packets) from flows belonging to specific applications and classifies P2P traffic using ML algorithm, namely the C4.5 decision tree. The datasets used involved TCP and UDP flows belonging to Skype, Thunder, PPTV, and non-P2P applications. The experimental results achieved for PPTV, Skype, and Thunder include precision values of 93.66, 91.01, and 90.96% and recall values of 92.82, 86.69, and 95.73 %, respectively.

Yang et al. [88] proposed a cocktail approach consisting of three sub-methods for identifying BitTorrent traffic. The first sub-method uses application signatures to identify unencrypted BitTorrent traffic. The second sub-method uses a message-based approach to perform identification of encrypted BitTorrent traffic. Here, after reassembling the

bidirectional flows into streams of messages, the length & direction of the first three messages were observed. If it satisfies specific criteria of message stream encryption (a protocol used to conceal traffic), the flow is classified as encrypted BitTorrent traffic. The third sub-method uses a signaling-based approach to perform pre-identification of BitTorrent traffic. Here, the prediction of BitTorrent flows takes place using the SYN flag of the first packet only. The authors evaluated their approach using modified Vuze clients, which generated real BitTorrent traffic and labeled the traffic themselves in benchmark traces. The experimental results achieved false positive, precision, and recall rates ranging from 1.31 to 2.47%, 98.26 to 99.03%, and 85 to 98%, respectively. This approach has the ability for real-time identification with low overhead.

2.4.2 Classification of encrypted traffic

Nowadays, due to the widespread use of encrypted communication for protecting personal information and/or concealing exchanged information, identification accuracy is dropping. For example, encryption is used in P2P file sharing, VoIP, and ISPs offering virtual private networks for communication. These factors reflect that encryption is going to increase, and it becomes difficult for network administrators to the traffic since its characteristics get changed when it is encrypted. Hence, various identification approaches classify encrypted traffic as either unknown traffic or wrongly interpret encrypted traffic as belonging to the same application, even though various encrypted applications may be mixed in traffic. Hence, most of the existing methods can be expected to become less effective. There exist some studies for addressing this issue that make use of modern classification techniques for P2P traffic classification, some of which are discussed below.

Korczynski and Duda [89] proposed stochastic fingerprints based on first-order homogeneous Markov chains to identify the various applications with encrypted traffic flows. They studied twelve representative applications (which includes Skype), whose parameters were identified by observing training application traces. Their technique achieved good accuracy as fingerprint parameters of applications differ considerably. The issue with this technique is that, as application fingerprints change over time, they need to be updated periodically. For the P2P application (Skype), the experimental results achieved a true-positive rate of 98.6 % and a false-positive rate of 0.1%.

Alshammari and Zincir [90] proposed a novel technique to identify VoIP encrypted traffic that is based on machine learning which generated robust signatures. They extracted

statistical features from network traffic flows without the use of information regarding payload or port numbers & IP addresses of source and destination. Three different sampling techniques (i.e., uniform random sampling, continuous data stream and stratified sampling) were studied on three ML algorithms (C5.0, Genetic programming and AdaBoost) that were trained on various training datasets. Here, uniform random sampling was found to be most appropriate for enhancing the automatic generation of robust signatures. Experimental results showed that C5.0 outperforms GP & AdaBoost algorithms while classifying multiple VoIP applications and classified Skype traffic with detection rate ranging from 80.3 to 99.6 % and false-positive rate ranging from 0.7 to 3.8%. But, for other network applications, this technique needs to be explored for its accuracy.

Kumano et al. [91] proposed a technique to identify real-time encrypted traffic. They focused on maintaining high accuracy by obtaining traffic features using a few packets only. They used two types of encryption (IPSec and PPTV) and employed two machine learning algorithms (C4.5 and SVM) for classifying the type of encryption and identification of application. Their work shows how accuracy degrades by reducing the number of packets and proposed a technique to identify traffic features with a few packets. By varying the number of features & packets, they achieved overall accuracy ranging from 79.3 to 92.5%. More packets can be reduced for some features by eliminating initialization packets, but detailed estimation & exploration are required to be done.

Wang et al. [92] proposed a novel approach based on the Hidden Markov Model for identifying network activities of encrypted traffic. In their technique, time series and statistical characteristics of packets are considered for analysis. Four-time series sequences during the interaction of four activities (session request, data transfer, response to session request, and response to data transfer) are analyzed for distinction, due to which packet inter-arrival time is considered as a feature. Similarly, due to distinction in packet sequences of four activities, packet length & packet inter-arrival time are selected as features for statistical characteristics. TeamViewer (which allows encrypted communication between hosts) is used to verify the efficiency of the approach. The datasets utilized include audio, video, transfer, and chat traffic types. Experimental results achieved a true-positive rate ranging from 96.4 to 99.1% and a maximum false-positive rate of 3.6%. However, unsupervised learning methods of modeling and further analysis of complex activities need to be considered further.

Du and Zhang [93] identified P2P traffic by utilizing the K-means algorithm that monitors flow information of TCP connections and calculates the distance. Their approach

focused on three TCP file-sharing P2P applications, namely BitTorrent, BitSpirit, and eMule. Experimental results achieved an average true-positive rate of 92.64, 96.22, and 99.76% for BitTorrent, BitSpirit, and eMule, respectively. The algorithm proposed by the authors is simple, feasible, slight overhead of time, and can be used for real-time detection of traffic.

Datta et al. [94] proposed a novel technique using application behavior-based feature extraction to detect Google-hangout traffic by taking it as a case study. Three machine algorithms were used, namely Naive Bayes, J48 decision tree, and AdaBoost, to classify traffic. The datasets consisted of traffic traces of Google-hangout, Gmail, and Google-plus, since these Google services share common behavior between them. The classification results had the recall values of 100% with J48 and AdaBoost separately and 99.98% with Naive Bayes.

By considering various traffic classification techniques (i.e., port-based, payload-based, and Classification in Dark) along with their advantages & limitations, Figure 2.3 compares them by considering their implementation, resource requirements, and performance in classifying traffic. Hence, the comparison factors include ease of implementation, requiring less computation, classification accuracy, classification of encrypted traffic, classification in real-time, and classification of unknown traffic. Each technique is given a value on a particular factor ranging from 1 to 3, where value 3 represents comparatively highest performing technique and value 1 represents comparatively lowest-performing technique. The port-based technique has the highest value while considering the factors of ease of implementation and less computation requirement. This technique can classify encrypted traffic and real-time classification, but it has the lowest value in all remaining factors (i.e., classification of encrypted traffic, classification in real-time, classification accuracy, and classify unknown traffic) since current generation P2P applications masquerade or utilizes random port-numbers due to which it will not give accurate results. Payload-based classification has the highest performance when classification accuracy is of prime importance. Due to this fact, it is widely used for ground truth verification of traffic; but comparatively, it doesn't perform well on other remaining factors. Classification in Dark has the highest performance while considering encrypted traffic classification, real-time classification, and unknown traffic classification.

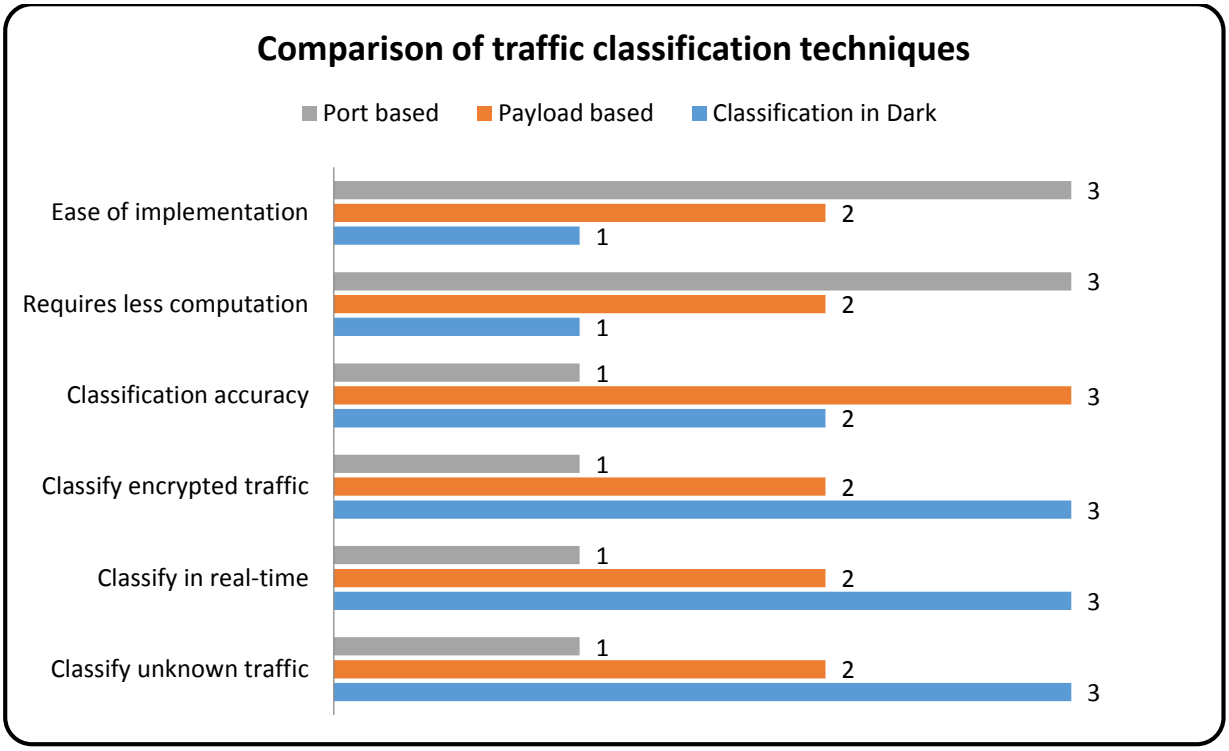


Figure 2.3. Comparison of traffic classification techniques based on their performance by considering various factors.

The comparison regarding various traffic classification techniques is referred to [18].

CHAPTER 3

CLASSIFICATION OF P2P NETWORK TRAFFIC

3.1 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. 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 to 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 [2]. 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 [3].

The main issue with P2P traffic is that it consumes a large amount of network bandwidth [2] [95] [96] [97]. Conventional network devices cannot handle the traffic of P2P applications, due to which network administrators and ISPs face various challenges such as providing excellent broadband experience to customers, purchasing of backbone links, and up-streaming bandwidth, which is costly. Considering the overall network traffic, which is composed of traffic from various application protocols (for example, SMTP, FTP, DNS, HTTP, P2P, HTTPS, etc.), traffic from P2P applications alone consumes a significant portion of the available 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 to avoid network congestion and maintain Quality of Service for various applications in their network, as shown in Figure 3.1.

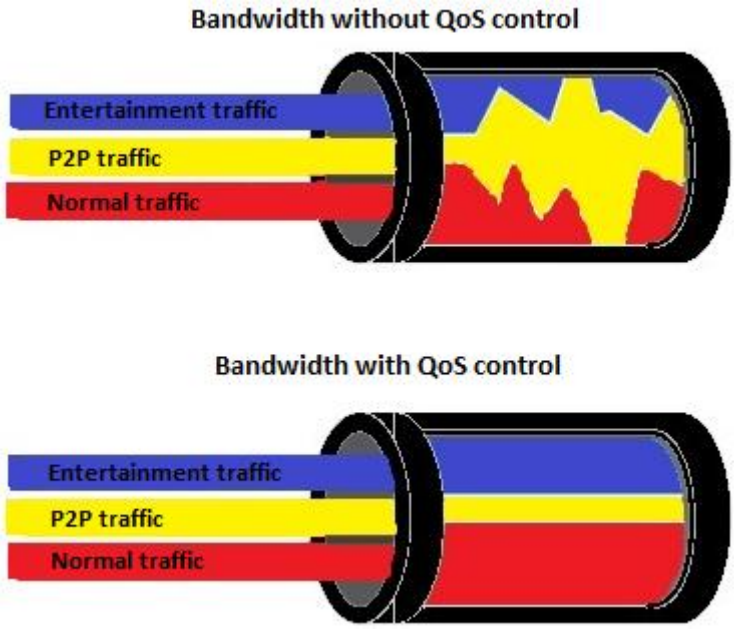


Figure 3.1. Controlling the quality of service.

Nowadays, classifying P2P traffic with high accuracy is a difficult task as various P2P applications either masquerade or encrypt their traffic to avoid detection [18] [98]. 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) [98] [29] [47]. 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 technique relies 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 [18]. 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 chapter, which is the combination of heuristic-based and statistical-based techniques.

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

- It has the ability to classify P2P traffic with high accuracy.

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

The experiments performed using the proposed hybrid technique achieved a high classification accuracy, 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 chapter perform equally with both TCP & UDP traffic flows and are not affected even if a traffic flow is encrypted.

The remaining chapter is organized as follows. Section 3.2 examines the related work. Section 3.3 discusses the multi-level P2P traffic classification technique. Section 3.4 discusses the evaluation criteria and experimental results. Finally, Section 3.5 provides the summary.

3.2 Related work

P2P applications have become very popular in 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 of the largest contributors to internet traffic [4], 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 [99]. Therefore, such kind of traffic can be observed naturally over 24 hours. 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. There exist various studies which either employ a hybrid approach or non-hybrid approach to classify P2P traffic.

Reddy and Hota [61] 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) [97] 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. [81] assessed four supervised and one un-supervised ML algorithms, namely SVM, C4.5 decision tree, Ripper, Naive Bayesian, and K-means, respectively, to identify P2P applications. They found that Ripper and C4.5 algorithms have 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 BitComet, BitTorrent, and uTorrent.

Tseng et al. [100] presented a methodology to classify P2P traffic based on 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. [101] utilized the Bat algorithm to search the most relevant parameters which can be used with SVM for classifying P2P traffic and were able to achieve the classification accuracy ranging between 86.77–91.34%.

Abdalla et al. [102] proposed a multi-stage method for feature selection to create a subset of optimal statistical traffic features that can be utilized for the online P2P traffic classification. 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. [103] 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. [104] 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 labeling 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 [85] [86] [105] proposed a hybrid technique to classify P2P traffic in two steps. The first step performs classification at the packet-level using the combination of 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 unidentified 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. [106] 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 employing 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 a 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.

Li et al. [107] proposed a hybrid classification technique using the combination of the 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. [108] 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 an accuracy of 95.67%, but it relies on dedicated hardware to classify P2P traffic.

Keralapura et al. [97] presented 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 [109] proposed a technique that uses the combination of pattern-based and statistical-based approaches to classify the traffic as P2P & non-P2P and achieved a maximum classification accuracy of 91.42%. The authors proposed another hybrid technique in [110], 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 chapter). 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 [85]. 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 high accuracy in classifying P2P traffic. Besides, 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 but rather utilizes a set of heuristic rules proposed in this chapter, which comparatively involves less computation, performs equally with both TCP & UDP traffic flows, and is not affected even if the traffic flow is encrypted.

3.3 Multi-level P2P Traffic Classification Technique

Based on the previous analysis, a multi-level P2P traffic classification technique is proposed. It is split 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.

3.3.1 System Model Assumptions

The proposed system model makes the following assumptions:

1) All packets of network traffic consist of IP-header and use either TCP or UDP protocol for communication. Therefore, all other packets in the dataset without IP-header are considered insignificant.

2) In a traffic flow, both source & destination peers transfer at least 100 bytes to each other. Therefore, small traffic flows where less than 100 bytes are transferred in both directions (i.e. from source to destination and vice-versa) are considered insignificant, so that such traffic flows are not misclassified as P2P traffic.

3.3.2 System Model for Classifying P2P Traffic

Figure 3.2 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 [111] and Weka [112].

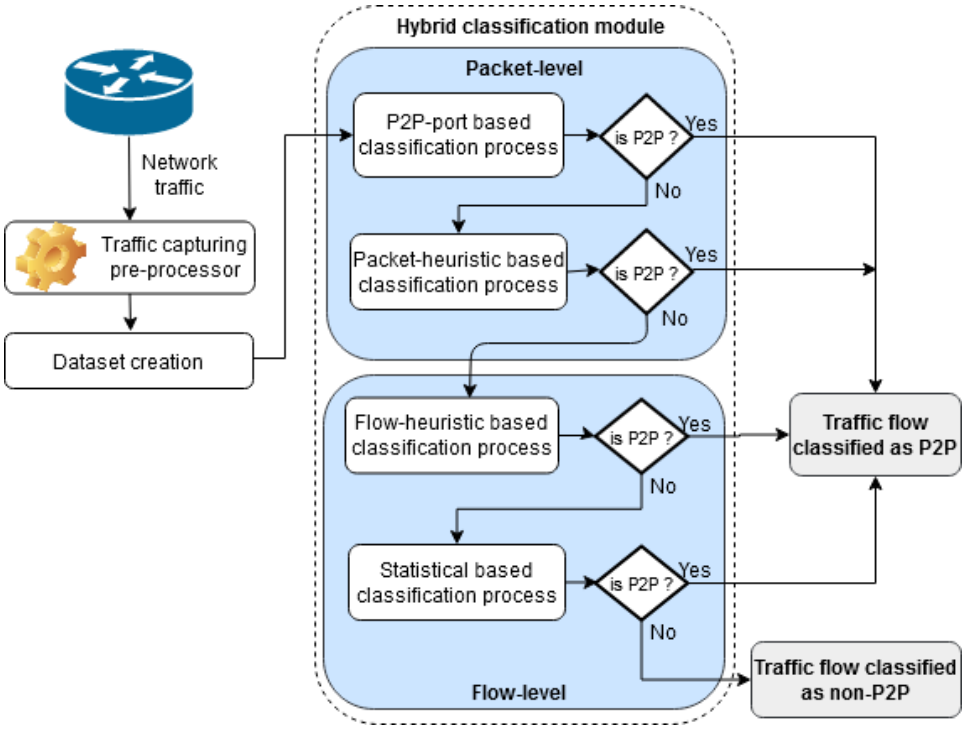


Figure 3.2. Multi-level P2P traffic classification technique.

While performing the task of traffic classification, a combination of five network parameters (i.e., source-IP, source-port, destination-IP, destination-port, and protocol) are generally used to define the traffic flow [51]. 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 by combining the five-tuple flow information, as shown in Figure 3.3. In this way, packets belonging to the same flow and traveling in either direction will have a similar 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.

```
if (srcPort > dstPort) then
        HashKey (packet) = "srcIP + srcPort + dstIP + dstPort + protocol"
else
        HashKey (packet) = "dstIP + dstPort + srcIP + srcPort + protocol"
```

Figure 3.3. Calculation of the packet hash-key.

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.

3.3.3 Packet-Level Classification Process (First Step)

As shown in Figure 3.2, 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 3.4. 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 3.3) is calculated and mapped with the information stored in the P2P flow table (which contains the records of the already classified P2P flows) 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 3.4). In Figure 3.4, while fetching the packets from the dataset, "is end?" checks whether the end of dataset (of packets) is reached or not.

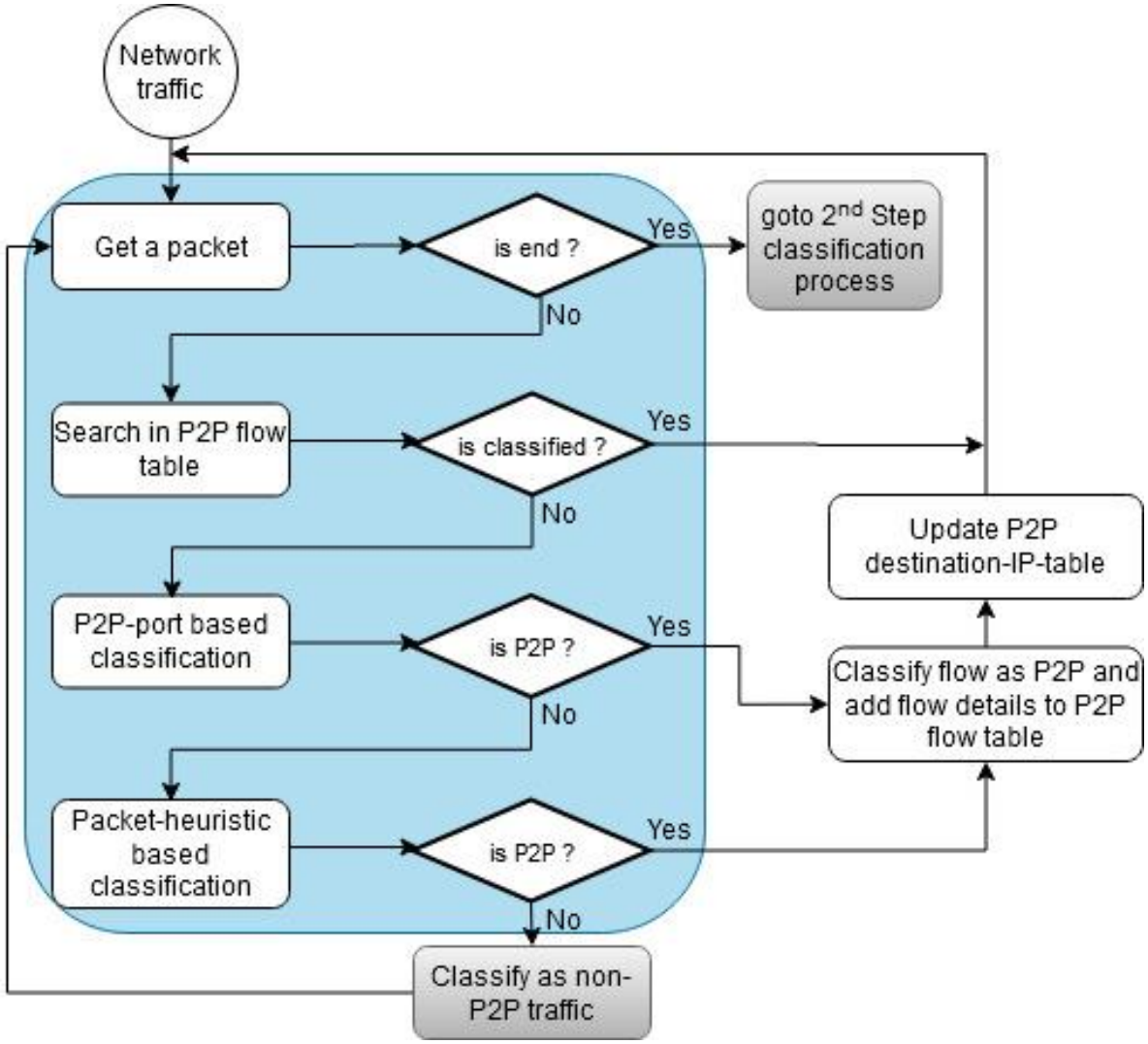


Figure 3.4. The packet-level classification process (First step).

3.3.3.1 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 (shown in Table 3.1) used by various P2P applications. If a match is found, then its flow is considered as P2P flow. Accordingly, the flow details are added in the P2P flow table, and the destination-IP-table is updated. Although it is well-known that the port-based technique is inefficient in traffic classification, it has been used here to perform an early classification of the P2P traffic, which may not be masquerading and still using well-known P2P port numbers [18] [98] for communication.

Table 3.1. List of well-known ports used by various peer-to-peer (P2P) protocols.

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

If α represents a P2P flow, then mathematically this classification process can be represented as shown in equation *(3.1)* below:

$$C1: f_i = \begin{cases} \alpha & \forall\ p \in d \\ \beta_1 & otherwise \end{cases} \qquad (3.1)$$

where, p = TCP/UDP port of a packet

n = total no. of packets

m = no. of traffic flows; where $m < n$

f_i = i^{th} traffic flow; where $1 <= i <= m$

d = database of well-known P2P port numbers

β_1 = non-P2P flow

The traffic flows which are classified as β_1 in equation (*3.1)* are used as an input in the next classification process.

3.3.3.2 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 based on 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:* It is well-known that an application makes use of the transport-layer port number to communicate over a network. 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 with the requested data using a well-known port number. Therefore, if the source & destination ports of a packet is 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. This heuristic can be represented as shown in equation *(3.2)* below:

$$H_1: \quad f_i = \alpha \quad if \; \exists \, (S_{port,i} \in L_e) \cap (D_{port,i} \in L_e) \qquad (3.2)$$

where α = P2P flow

f_i = i^{th} traffic flow; where $1 <= i <= m$

$S_{port,i}$ = source-port of packet belonging to a flow f_i

$D_{port,i}$ = destination-port of packet belonging to a flow f_i

L_e = list of ephemeral ports i.e. [1024 – 65535]

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 & 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 [61] [113]. For example, a Skype peer communicates with the super-peer using both TCP and

UDP protocols. Therefore, if a source-IP uses TCP & 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 [97]. This heuristic can be represented as shown in equation *(3.3)* below:

$$H_2: \quad f_i = \alpha \quad if \; \exists \, (I_{src,i} \in P_{tcp}) \cap (I_{src,i} \in P_{udp}) \tag{3.3}$$

where $I_{src,i}$ = Source-peer IP packets of a flow f_i

P_{tcp} = TCP protocol

P_{udp} = UDP protocol

3) *Communication with destination-IP, which is already classified as P2P:* Before the communication between peers, a peer waits for the incoming connections from the other peers with the help of a listening port [44]. Figure 3.5 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 communicates 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. This heuristic can be represented as shown in equation *(3.4)* below:

$$H_3: \quad f_i = \alpha \quad if \; \exists \, (S_{ip,i} \in T_{ip}) \cap (D_{ip,i} \in T_{ip}) \tag{3.4}$$

where T_{ip} = table storing < IP, port > pair information of those peers which are already classified as P2P.

$S_{ip,i}$ = < IP, port > pair information of source-peer belonging to flow f_i.

$D_{ip,i}$ = < IP, port > pair information of destination-peer belonging to flow f_i.

4) *Usage of consecutive port numbers:* It has been observed that various P2P applications actively make many 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) [114]. Figure 3.6 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. If we consider that:

k = no. of ports used by a source peer.

p_r = initial (random) port number used by a source peer for communication.

$x = \{ p_{r+1}, p_{r+2}, \ldots, p_{r+k} \}$

$L = \{ 1, 2, 3, \ldots, k \}$

$y = \{ x[j] - p_r, x[j+1] - p_r, \ldots x[j+k] - p_r \}$; $0 <= j <= k$

then, this heuristic can be represented as shown in equation *(3.5)* below:

$$H_4: \; f_i = \alpha \quad if \; \exists \; c \mid \; count(y \cap L) >= c; \qquad c=3 \tag{3.5}$$

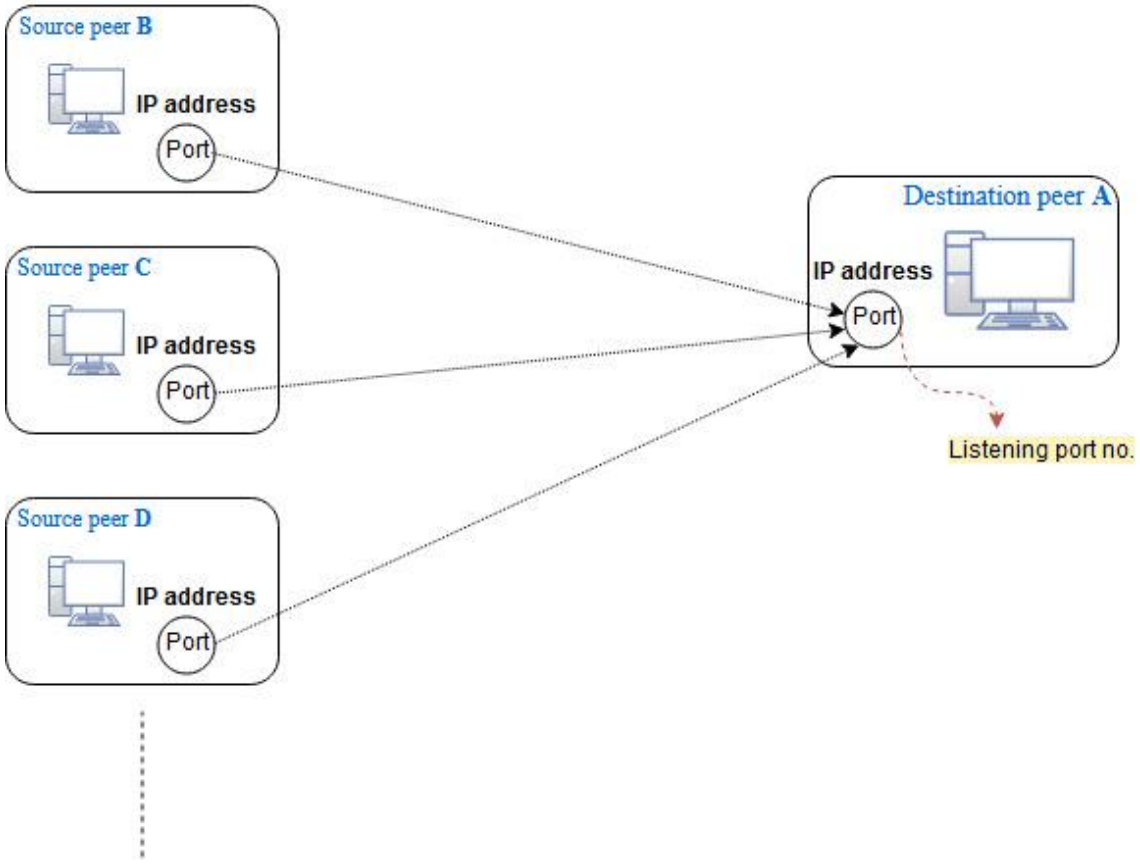


Figure 3.5. Connection pattern of source peers with the destination P2P peer.

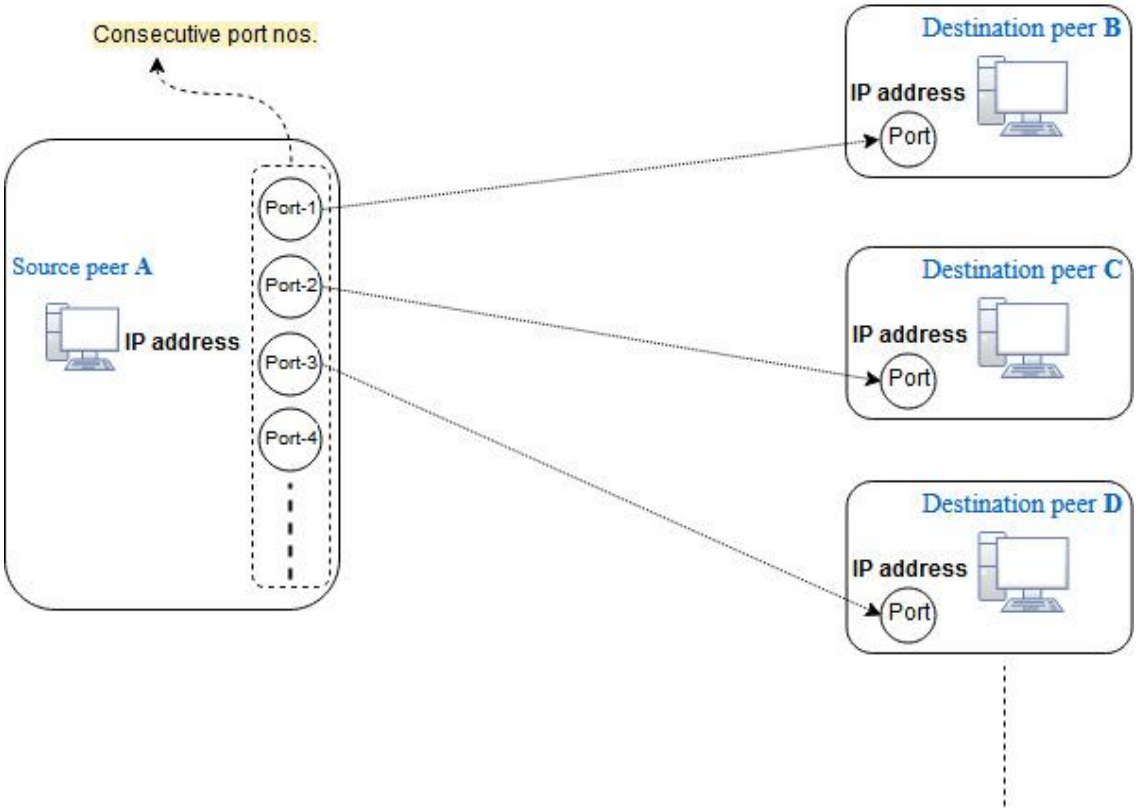


Figure 3.6. Connection pattern of source P2P peer with the destination peers.

Generally, a heuristic process can be represented as shown in equation *(3.6)* below:

$$H_i = \begin{cases} True & if\ \exists\ f_i \in \alpha; \quad 1<=i<=4 \\ False & otherwise \end{cases} \qquad (3.6)$$

So overall, the packet heuristic process can be represented as shown in equation *(3.7)* below:

$$C2:\ f_i = \begin{cases} \alpha & (H_1 \cup H_2 \cup H_3 \cup H_4) = True \\ \beta_2 & otherwise \end{cases} \qquad (3.7)$$

The traffic flows which are classified as β_2 in equation (*3.7)* are used as an input in the next classification process.

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 & UDP traffic and are not affected even if the traffic is encrypted. Table 3.2 shows Algorithm-3.1 that performs packet-level classification process and classifies P2P traffic in the First step. The traffic which remains un-classified as P2P is fed to the flow-level classification process (i.e., Second step).

Table 3.2. Algorithm for performing Packet-level traffic classification.

Algorithm-3.1: 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

3.3.4 Flow-Level Classification Process (Second Step)

Figure 3.7 shows the flow-level classification process, which is the combination of flow-heuristic-based and statistical-based techniques.

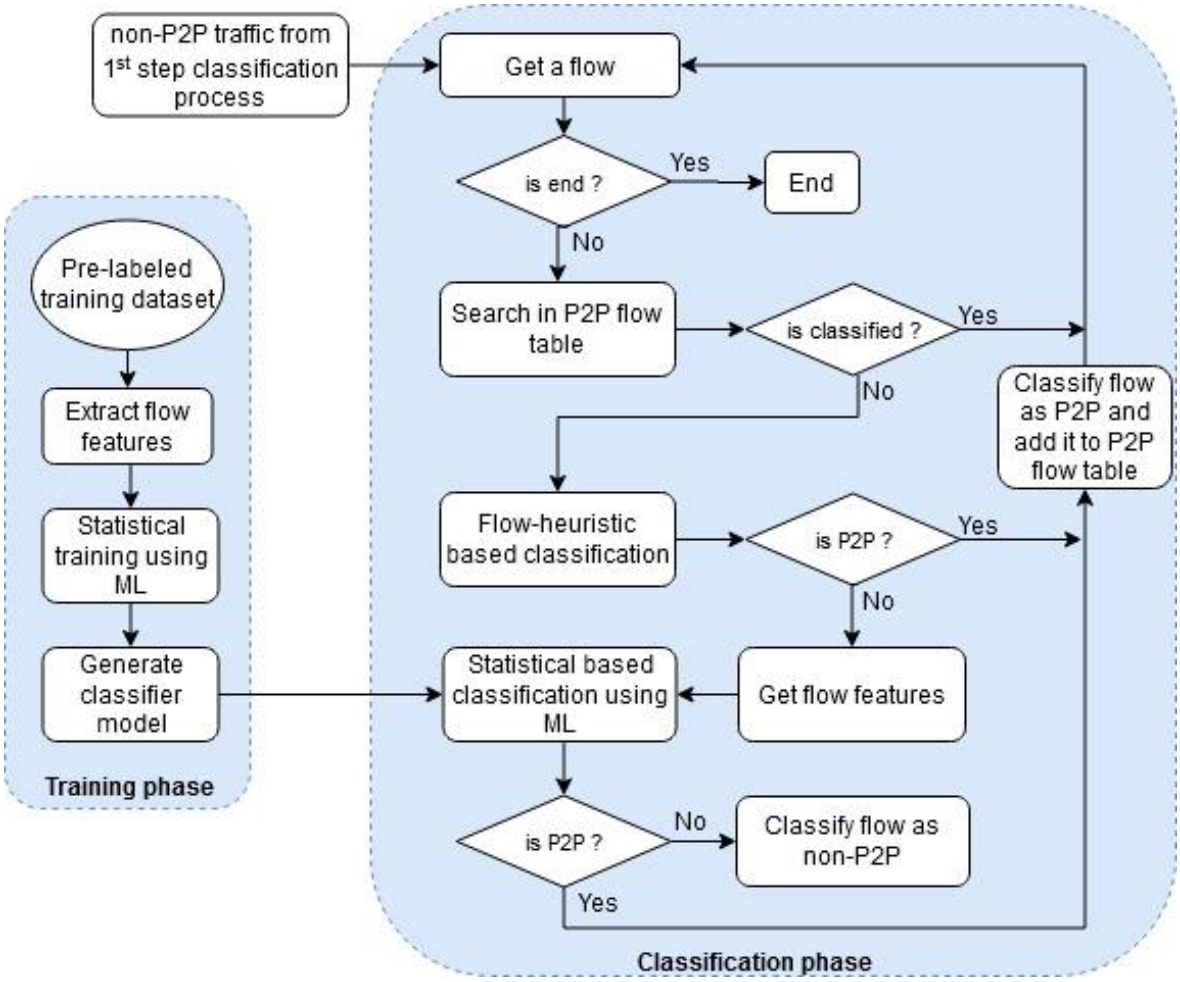


Figure 3.7. The flow-level classification process (Second step).

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) to verify whether it is already classified as P2P flow or not. In Figure 3.7, while fetching the traffic flows from the dataset, "is end?" checks whether the end of dataset (of flows) is reached or not. The flows which are not classified as P2P are fed to the flow-heuristic-based classification process, which is explained below.

3.3.4.1 Flow-Heuristic Based Classification

One of the P2P application properties is that it behaves as both a client & server simultaneously, i.e., data is transferred from destination to source and source to destination at the same time. 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. The Flow heuristic process can be represented as shown in equation *(3.8)* below:

$$C3: f_i = \begin{cases} \alpha & \text{if } \exists \ (t_s >= T) \cap (t_d >= T) \ ; \quad T>=3 \\ \beta_3 & \text{otherwise} \end{cases} \qquad (3.8)$$

where β_3 = non-P2P flow

t_s = data transferred from source to destination in a flow f_i.

t_d = data transferred by destination to source in a flow f_i.

T = threshold amount of data transferred (in MB).

The traffic flows which are classified as β_3 in equation (*3.8)* are used as an input in the next classification process.

3.3.4.2 Statistical Based Classification

The traffic-flows, which 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 ML algorithm, namely the C4.5 decision tree to classify the remaining traffic (as shown in Figure 3.7). This process involves the training phase as well as the classification phase. A classification model is created using the training dataset, which contains both P2P & non-P2P traffic flows in the training phase. The ML algorithm analyses the relationship between the flow features and the output class value to generate a classifier model, which predicts the type of traffic flow by analyzing 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 match the distinct characteristics of P2P traffic, then the flow is considered as P2P.

C4.5 decision tree uses the mathematical model as shown in equation *(3.9)* to build a classification tree (by calculating & deciding the root node):

$$\text{C4:}\quad Gain(S) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Entropy(S_i) \qquad (3.9)$$

$$\text{and}\quad Entropy(S) = \sum_{i=1}^{n} -p_i \log_2 p_i \qquad (3.10)$$

where S = the set of cases

n = number of partitions

$|Si|$ = number of cases in the partition i.

$|S|$ = number of cases in S.

p_i = proportion of S_i to S.

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 [85], as well. They are given as input to the ML algorithm to build a statistical-based classifier for performing the classification. The C4.5 decision tree is chosen as the ML algorithm for traffic classification since it is faster and better compared to other ML algorithms [36]. Algorithm-3.2 in Table 3.3 shows the flow-level classification process (i.e., Second step), which performs P2P traffic classification at the flow level.

Table 3.3. Algorithm for performing Flow-level traffic classification.

Algorithm-3.2: 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
ft: 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( ft.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

Using equations *(3.1), (3.7) (3.8)* & *(3.9)*, overall classification of the model can be represented as shown in equation *(3.11)* below:

$$C = C1 \cup C2 \cup C3 \cup C4 \qquad (3.11)$$

3.4 Verification

The implementation process is accomplished using Java programming language along with a java API named jNetPcap [111], which is used to read the network packets for extracting various statistical features such as packet length, port number, IP-address, number of bytes sent or received, etc. Initially, various P2P & non-P2P applications are executed to generate network traffic. As shown in Figure 3.2 (in the previous section), the network packets are then captured from a terminal node of a network with the help of a packet-capturing tool called Wireshark [16] to create a dataset, which is in the form ".pcap" trace file. Packets are read from the trace file using the jNetPcap library and fed into the classification model.

Initially, the packet-level classification process (as mentioned in Algorithm-3.1 in Table 3.2) is utilized where P2P-port-based classification is performed by extracting TCP/UDP port number from each packet header and mapped with the database of well-known P2P port numbers that most of the P2P applications may use while communicating. The packets which remain unclassified as P2P then undergo analysis using packet-level heuristics where a set of heuristic rules is used to classify the traffic. The traffic which remains unclassified as P2P undergoes further analysis in the flow-level classification process (as mentioned in Algorithm-3.2 in Table 3.3), where traffic is analyzed & classified using a flow-level heuristic. Finally, the remaining unclassified traffic undergoes analysis using the statistical-based technique, where ML algorithm C4.5 decision tree is utilized for classification purpose. Here, the ML algorithm is trained using various statistical properties of traffic flow (as mentioned in the previous section). It generates a classification model which is then used to classify the traffic either as P2P or non-P2P. For performing statistical classification with the C4.5 decision tree, an open-source library known as Weka [112] is utilized, which contains the collection of various ML algorithms.

3.4.1 Complexity Analysis

The complexity of the proposed technique is analyzed with the help of asymptotic notation. Let "n" is the total no. of packets processed by the proposed system model; then the complexity of every process in the system is represented as follows:

$T_1(n)$ → P2P-port-based process

$T_2(n)$ → Packet-based heuristic process

$T_3(n)$ → Flow-based heuristic process

$T_4(n)$ → Statistical-based process

We assume that "c1" is the constant time taken to process each packet by P2P-port-based process, then its complexity is:

$T_1(n) = n*c1 \rightarrow O(n)$

We assume that "c2" is the constant time taken to process all packets by each heuristic in the Packet-heuristic-based process, then the complexity of all four heuristics is:

$T_2(n) = 4*n*c2 \rightarrow O(n)$

We assume that "c3" is the constant time taken to process each packet by Flow-heuristic-based process, then its complexity is:

$T_3(n) = n*c3 \rightarrow O(n)$

We assume that "k" is the number of features used in the ML algorithm (which is constant in our system model), then the complexity of the C4.5 decision tree algorithm [115] used in Statistical-based process is:

$T_4(n) = k*n*\log(n) \rightarrow O(n*\log(n))$

Hence, overall complexity of the proposed classification technique is:

$T(n) = T_1 + T_2 + T_3 + T_4 = O(n) + O(n) + O(n) + O(n*\log(n)) = O(n*\log(n))$

3.4.2 Evaluation Metrics

The performance of a classifier can be characterized with the help of metrics known as False Positive (FP), True Positive (TP), False Negative (FN), and True Negative (TN). They are described as follows:

1) TP: Percentage of instances correctly categorized as associated with a specific class.
2) TN: Percentage of instances correctly categorized as not associated with a specific class.
3) FP: Percentage of instances incorrectly categorized as associated with a specific class.
4) FN: Percentage of instances incorrectly categorized as not associated with a specific class.

The proposed technique classifies the traffic flow as P2P or non-P2P. Accuracy, Recall, and Precision metrics are employed 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 in equations (*3.12*), (*3.13*), and (*3.14*).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.12}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.13}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.14}$$

3.4.3 Datasets, Validation, and Experimental Results

The proposed technique is evaluated using two offline traffic datasets, which are realistic, and consist of both P2P and non-P2P flows, as shown in Table 3.4.

Table 3.4. The number of flows in the datasets.

Dataset	P2P (no. of flows)	non-P2P (no. of flows)	Total
Dataset-1	20,617	48,179	68,796
Dataset-2	3881	2892	6773

The first traffic dataset (i.e., Dataset-1) is UNIBS [32] [116] 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 [16] tool, and their pattern of communication was observed. Therefore, the flows which belong to the P2P traffic are well-known in advance. Besides, such traffic flows are labeled accordingly with actual applications for 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 3.5.

Table 3.5. Summary of the collected data.

Protocol	Packets	Bytes
POP3	13,647	918,878
IMAP	3191	213,554
HTTP	1,399,230	92,060,704
BitTorrent	379,836	329,477,265
SSH	2,586,027	141,334,606
RTMP	11,712	779,616
Dropbox	6498	429,308
StarCraft	7	394
FTP_CONTROL	19	1274
Telnet	90	6132
SOCKS	2487	139,650
Skype	30	3657
Others	402,357	26,674,298

We made the training and testing dataset by combining both datasets, as shown in Table 3.4. In the statistical-based classification process, the datasets were split into training & 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 encrypted P2P traffic to evaluate 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 3.8), which also show that it can classify the encrypted traffic.

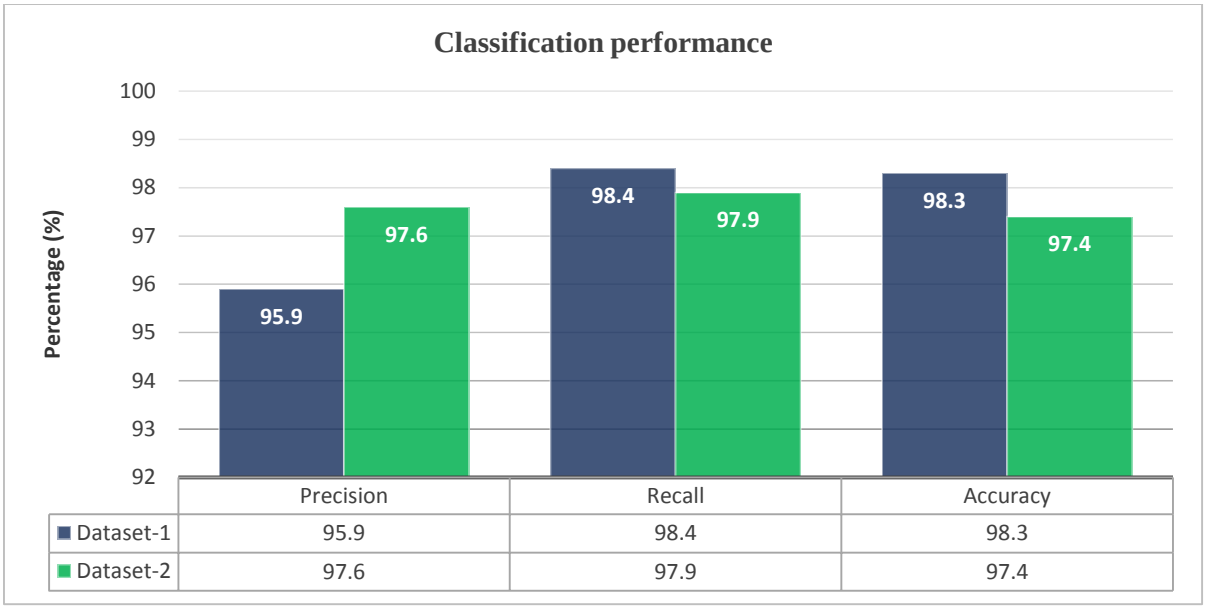


Figure 3.8. Classification performance of the proposed hybrid technique.

In the proposed hybrid technique, after analyzing the type of protocol (i.e., TCP or UDP) used by packets for a 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 ML algorithm, namely: C4.5 decision tree) to perform the classification, which is independent of traffic using the TCP or UDP protocol for communication. Hence, the overall proposed hybrid technique can work with both TCP and UDP protocols at every step. Besides, 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 [18].

Furthermore, the classification performance of the proposed hybrid technique at various stages is shown in Table 3.6. 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 3.6. Classification performance at various steps (P → P2P-port-based, PH → packet-heuristic-based, FH → flow-heuristic-based, S → statistical-based).

Classification Process	Accuracy (%)
P	11.90
P + PH	90.50
P + PH + FH	95.10
P + PH + FH + S	98.30

In Table 3.6, 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 [18] [98] 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 based on 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.

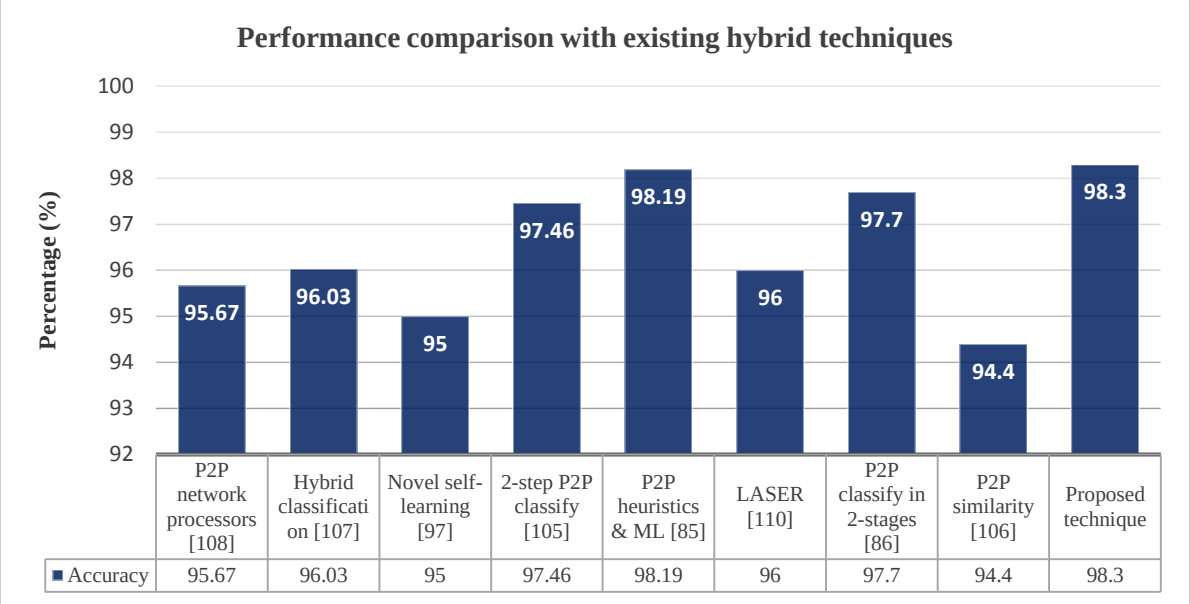


Figure 3.9. Accuracy comparison of various hybrid P2P traffic classification techniques.

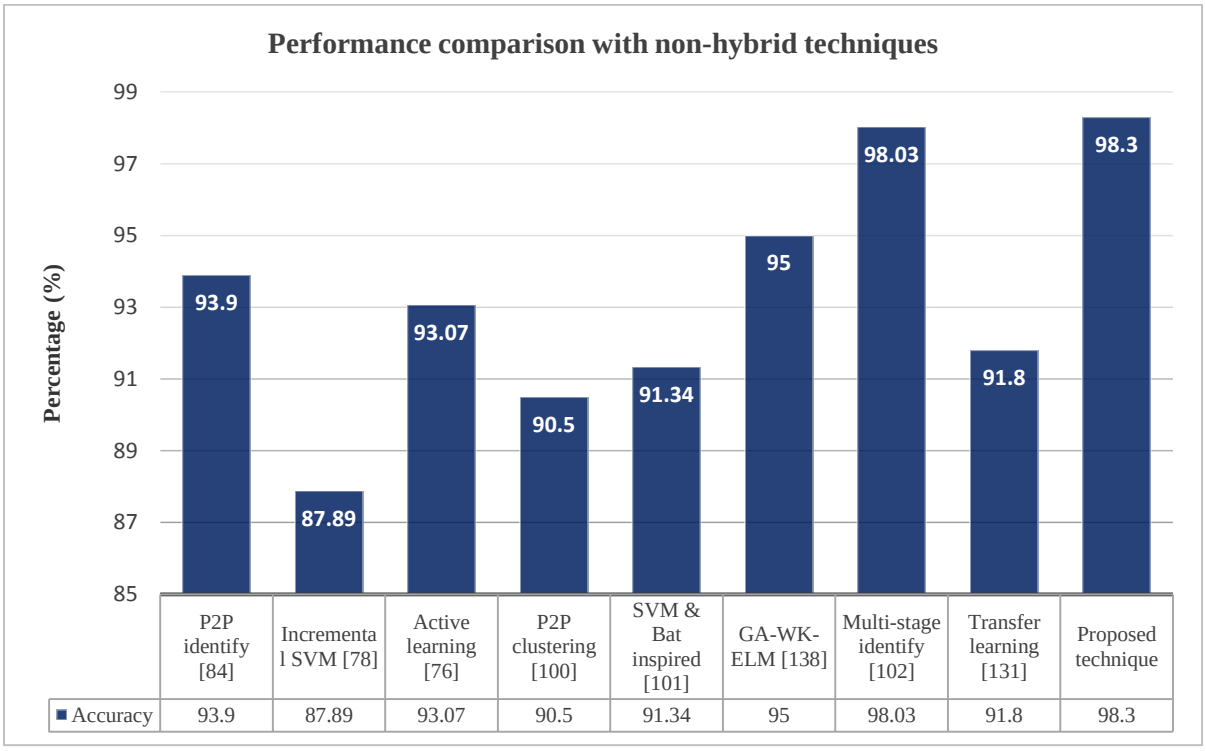


Figure 3.10. Accuracy comparison of proposed hybrid technique with existing non-hybrid techniques.

We compared our proposed hybrid P2P traffic classification technique with existing hybrid & non-hybrid techniques as well. Figure 3.9 and Figure 3.10 show that the classification accuracy achieved by our proposed hybrid technique is higher than the various existing hybrid as well as non-hybrid P2P traffic classification techniques. As mentioned in the previous section, there exist various metrics (such as recall, precision, accuracy, etc.) which can be used

to evaluate the performance of a traffic classification technique. Various authors used different metrics to evaluate their classification technique. So, for comparing the performance of our proposed technique with existing hybrid & non-hybrid techniques, the accuracy metric is chosen because it is the only common metric among all which is used for evaluating the performance of the P2P traffic classification technique. Here, all the hybrid & non-hybrid techniques being compared considers most of the popular P2P applications (such as Skype, BitTorrent, eMule, etc.) for classification purpose, which our proposed technique is also considering. Apart from the achieved accuracy, the proposed hybrid technique addresses the limitations of existing hybrid & non-hybrid techniques as well.

The proposed hybrid technique is compared with non-hybrid techniques on the basis that every single (non-hybrid) modern classification technique has its own limitation. For example, the heuristic-based technique has a limitation that the classification models built using this approach lose accuracy if applications evolve or change their communication patterns/behavior over time. On the other hand, the ML-based classification technique has a limitation that the accuracy of the classification results completely relies on the accuracy & quality of given training datasets upon which the ML algorithms rely to perform traffic classification. Also, it becomes almost difficult to accurately classify two or more traffic classes if they share similar characteristic features since in such a case, the traffic features of such classes will overlap with each other leading to the ML algorithm getting biased towards one of the traffic classes, which leads to inaccurate results. Therefore, a (single) non-hybrid technique may not be sufficient in classifying P2P traffic accurately, because depending upon the non-hybrid approach used to classify P2P traffic, it may not be applicable for real-time classification (e.g. DPI approach, which involves large computation) or may not be able to classify newer/proprietary application protocols [85]. Therefore, we consider our hybrid technique better than a single (non-hybrid) technique because it combines the benefits of modern classification techniques; such that if some traffic flow evades detection in one of the classification techniques, then it gets detected by other classification technique.

Further, the proposed hybrid technique is compared with existing hybrid techniques on the aspects other than achieved accuracy also, where it addresses their limitations, thereby making it better in comparison to them. Table 3.7 shows the comparative analysis of various existing hybrid classification techniques with our proposed technique. It specifies the techniques (i.e. port-based, signature-based, statistical-based, machine learning, heuristic-based) utilized by them in their approach for classifying P2P traffic and ML algorithm used for this purpose.

Further, it also specifies their applicability on other aspects, for example, whether they classify TCP/UDP/encrypted traffic, use dedicated hardware, and overall accuracy achieved.

Table 3.7. Comparison of hybrid P2P traffic classification techniques specifying the classification technique used/applicability which includes: port (Port), signature (Sign), statistical (Stat), machine learning (Mach), heuristic (Heu), ML-algorithm (Algo), use dedicated-hardware (Ded-hd), classify-tcp (TCP), classify-udp (UDP), encryption (Enc), accuracy (Acc).

Ref	Studies	Technique					Algo	Ded-hd	TCP	UDP	Enc	Acc (%)
		Port	Sign	Stat	Mach	Heu						
[108]	Chen et al. (2009)	✓	✓		✓	✓	Flexible neural tree-based	✓	✓	✓	✓	95.67
[107]	Li et al. (2009)	✓	✓		✓		C4.5 decision tree		✓	--	✓	96.03
[97]	Keralapura et al. (2010)		✓			✓			✓	--		95.00
[105]	Ye and Cho (2013)		✓		✓	✓	C4.5 decision tree		✓		✓	97.46
[85]	Ye and Cho (2014)		✓		✓	✓	REPTree		✓		✓	98.19
[110]	Sajeev and Nair (2016)		✓		✓		C4.5 decision tree		✓	✓	--	96.00
[86]	Ye and Cho (2017)		✓		✓	✓	REPTree		✓	--	--	97.70
[106]	Khan et al. (2019)	✓			✓		Decision tree		✓	--	--	94.40
	Proposed hybrid technique	✓		✓	✓	✓	C4.5 decision tree		✓	✓	✓	98.30

During the classification process, the techniques used in [85] [86] [97] [105] [107] [110] rely on the signature-based approach, which is computationally expensive [18] [50] [76] and has various other limitations as discussed in Chapter 2. Besides, the techniques in [85] [86] [105] do not classify the UDP traffic. The technique used in [108] relies on dedicated hardware for the P2P classification, whereas the technique used in [106] 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 chapter 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 [18] [50] [76], but rather relies on heuristic-based and statistical-based techniques, which are comparatively light on resources. Besides, the proposed hybrid technique works with both TCP and UDP traffic flows and classifies the encrypted traffic, as well.

3.5 Summary

P2P applications have been used extensively 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 & 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 for maintaining Quality of Service for various applications in their network.

In this chapter, we proposed the multi-level P2P traffic classification technique, which is sub-divided into the packet-level and flow-level classification processes. By analyzing the behavior of various P2P applications, some heuristic rules have been proposed for classifying P2P traffic and are utilized in both the packet-level and flow-level classification processes. 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 ML algorithm to classify traffic as P2P or non-P2P. The experiments performed using the proposed hybrid technique achieved a high classification accuracy of 98.30%, which is greater 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. Besides, it also works with both TCP & UDP traffic and is not affected even if the traffic is encrypted. Therefore, the proposed hybrid technique aimed to address the following tasks using a single classification model, in comparison to existing P2P classification techniques:

- 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.

CHAPTER 4

CLASSIFICATION OF NETWORK TRAFFIC GENERATED BY P2P WEB-SERVICES INCORPORATING FILE-DOWNLOADS

4.1 Introduction

P2P technology has grown rapidly over the past decade and has been one of the largest contributors to internet traffic [85]. It has changed the composition of internet traffic from asymmetric to symmetric. The traffic such as WWW, FTP, Email, etc., uses the client-server approach where the client requests data from the server and the server provides the data back to the client, which leads to asymmetric kind of traffic and does not consume large bandwidth. However, with the advent and popularity of various P2P applications, the peers involved in communication acts as client & server simultaneously due to which transfer of data takes place in both directions in huge amount, which leads to symmetric-kind of network traffic and consumes a lot of network bandwidth. Therefore, non-P2P application protocols such WWW, FTP, Email, HTTPS, etc., are unable to get their fair share of network bandwidth and suffer from poor Quality of Service.

P2P file-sharing applications such as eMule, BitTorrent, etc., are the largest contributor of P2P traffic, which accounts for more than 44% of the internet traffic in EMEA (Europe, the Middle East, and Africa) [117]. As of late, the action taken against the utilization of P2P applications has constrained them to discover alternative ways (such as port disguising, traffic tunneling, traffic encryption, and so forth) to stow away their traffic on the internet. While the authors in [118] argue that P2P traffic is diminishing, yet it represents a major portion of the Internet traffic and is witnessed as the burden over the network resources [119]. Due to the massive adoption of P2P applications by the users in the past few years, it has attracted great attention of the network administrators and the ISPs as it poses various commercial problems such as network security, congestion, etc.

This chapter focuses on classifying P2P file-sharing (P2P-fs) traffic in the network. We specifically focus on P2P file-sharing traffic since it is the largest contributor to P2P internet traffic as a whole. For this purpose, a 2-step classification approach has been employed, which is categorized into packet-level and flow-level classification modules. In packet-level classification, the P2P-port-based technique has been utilized, and in the flow-level

classification, a combination of heuristic-based and statistical-based techniques (with ML algorithm, namely: C4.5 decision tree) has been utilized for classifying the traffic.

The remaining chapter is organized as follows. In Section 4.2, related work has been discussed. The proposed traffic classification methodology has been discussed in Section 4.3. In Section 4.4, traffic datasets, validation, and experimental results have been discussed. Finally, Section 4.5 presents the summary.

4.2 Related work

Park et al. in [120] identified the peers that use the popular BitTorrent client program known as uTorrent. The authors proposed a methodology to identify copyrighted file sharers by analyzing request-response packets of HTTP & UDP protocols as well as the handshake process of the BitTorrent client program. But, their methodology focuses only on identifying peers using a single file sharing program (i.e., uTorrent) and did not address the traffic encryption issue as well.

Reddy and Hota in [61] proposed a methodology to classify P2P traffic based on host behavior. They identified a set of heuristics that analyses the host behavior from the headers of the transport layer and found the average detection rate of 99%.

Ye and Cho in [85] used a hybrid technique, which is the combination of packet-level and flow-level classification processes, to classify P2P traffic. At the packet-level, the traffic is classified by combining signature-based & heuristic-based techniques. At the flow-level, a combination of statistical-based & pattern-heuristic-based techniques is used to classify remaining unknown traffic. The authors achieved a classification accuracy of 98%, but their technique does not classify P2P traffic, which uses UDP protocol for communication.

Jamil et al. in [103] proposed a technique that utilizes the combination of SNORT and ML-based techniques to classify P2P traffic. The technique used Chi-Square and fuzzy as feature selection algorithms along with ML algorithms: SVM, C4.5 decision tree, and ANN and achieved a classification accuracy of 99.5%.

Abdalla et al. in [102] proposed an approach based on feature selection and analytical methods (scatter & ANOVA) for detecting the optimal set of flow features that could be used for online P2P traffic classification. Their methodology used a four-stage process to narrow down and identify the optimal features for traffic classification. The ML algorithms used with

the selected features were J48 and Naïve Bayes, for which they achieved a classification accuracy of 99.5%.

Sajeev and Lekshmi in [110] proposed a hybrid technique that uses header & payload information to classify P2P traffic. In the first step, the communication module is created by analyzing header information, and then LASER, i.e., Longest Common Subsequence (LCS)-based Application Signature ExtRaction algorithm, is used for extracting signatures from the payload. Both the header and payload information are fed into a statistical-based classifier that uses the C4.5 decision tree to classify P2P traffic. The technique was able to achieve a detection rate of 95%, but since it also makes use of signature-based technique, therefore it may not be able to achieve good classification accuracy where the whole traffic is encrypted.

Most of the studies discussed above generally classify traffic as either P2P or non-P2P. The technique proposed in this chapter focuses on classifying P2P file-sharing traffic specifically, as it is one of the largest contributors to P2P internet traffic as a whole. For this purpose, a combination of heuristic-based and statistical-based classification approach (with ML algorithm, namely: C4.5 decision tree) has been utilized to classify whether a flow belongs to P2P file-sharing traffic or not.

4.3 P2P-fs Traffic Classification Technique

Based on the previous analysis, a multi-level P2P-fs traffic classification technique is proposed. It is split 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.

4.3.1 System Model Assumptions

The proposed system model makes the following assumptions:

1) All packets of network traffic consist of IP-header and use either TCP or UDP protocol for communication. Therefore, all other packets in the dataset without IP-header are considered insignificant.

2) In a traffic flow, both source & destination peers transfer at least 100 bytes to each other. Therefore, small traffic flows where less than 100 bytes are transferred in both directions (i.e. from source to destination and vice-versa) are considered insignificant, so that such traffic flows are not misclassified as P2P-fs traffic.

3) In addition to file-sharing functionality, P2P-fs applications may possess other functionalities also like chat, streaming, etc. Therefore, the proposed model assumes that P2P-fs applications generate only file-sharing traffic.

4.3.2 System Model for Classifying P2P-fs Traffic

A 2-step classification approach has been employed to classify P2P-fs traffic, as shown in Figure 4.1. The classification process has been categorized into two levels, namely packet-level & flow-level classification. In the packet-level classification process, P2P-port based technique has been utilized, and in the flow-level classification process, a combination of heuristic-based and statistical-based techniques (with ML algorithm, namely: C4.5 decision tree) has been utilized to classify the traffic either as P2P-fs or non-P2P-fs.

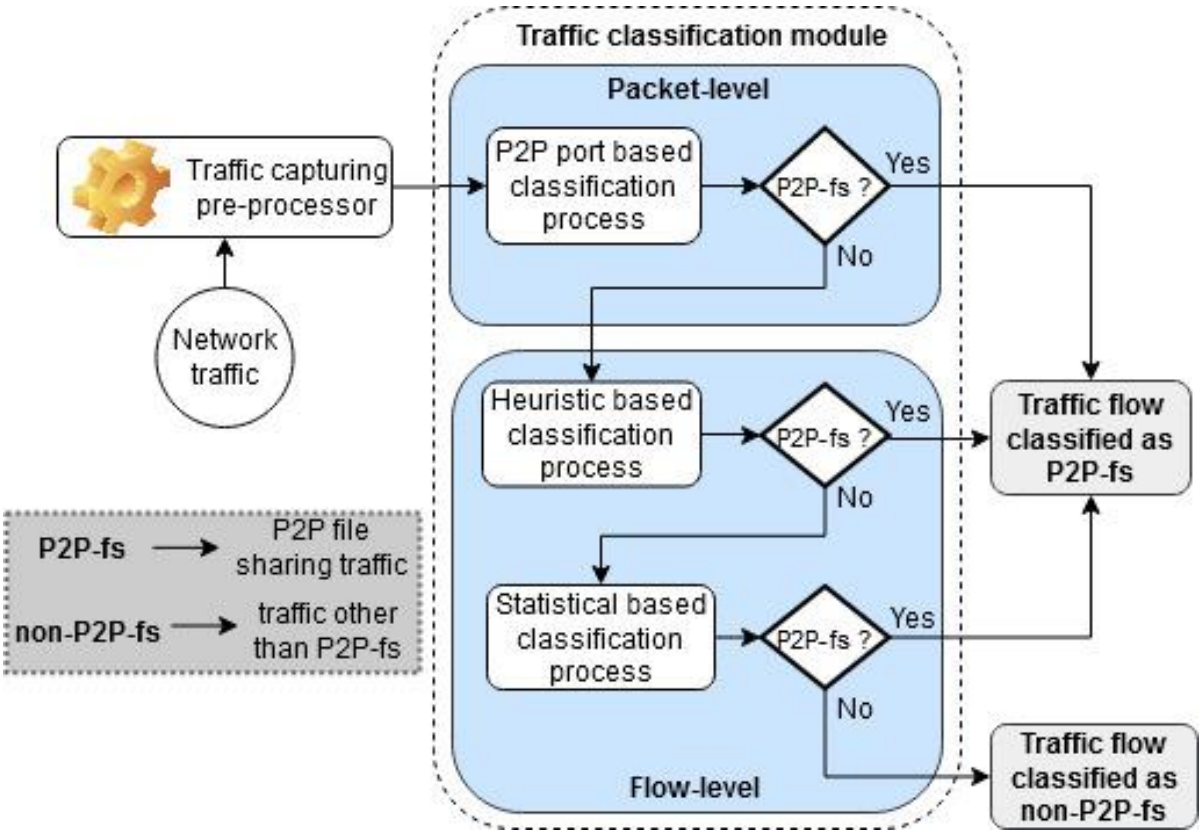


Figure 4.1. P2P (file-sharing) traffic classification technique.

In the traffic classification process, a flow is generally defined as the combination of 5-tuples (i.e., source-IP, source port, destination-IP, destination port, protocol). In order to identify packets belonging to the same flow, packet-hash is calculated by concatenating 5-tuple flow information as shown in Figure 4.2.

```
if (srcPort > dstPort) then
        HashKey (packet) = "srcIP + srcPort + dstIP + dstPort + protocol"
else
        HashKey (packet) = "dstIP + dstPort + srcIP + srcPort + protocol"
```

Figure 4.2. Calculation of hash-key of a packet.

In this way, packets of a flow that are traveling in either direction will have the same packet-hash. It is used to identify those packets whose flow has already been classified as P2P-fs. In this research work, a P2P flow table is used to store both the flow details and packet-hash of the flows, which have been classified as P2P-fs.

4.3.3 Packet-Level Classification Process (First Step)

As shown in Figure 4.1, 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.

4.3.3.1 P2P-Port Based Classification

In the packet-level classification process, initially, P2P-port based technique is employed by extracting the port number from the packet header and mapped with well-known P2P port numbers which may be used by various P2P-fs applications [121], as shown in Table 4.1.

Table 4.1. Default port numbers used by various P2P-fs applications.

P2P-fs applications	Default ports used
eMule	4662, 4672, 4711
uTorrent	771, 6881-6889, 6890, 6891-6900, 6901, 6902-6968, 6969, 6970-6999, 7000, 9091
Transmission	9091
qBitTorrent	9000

If a match is found, then the packets belonging to a particular flow are labeled as P2P-fs, and the flow details are added in the P2P flow table. Although the port-based technique is inefficient in classifying the traffic (since many P2P applications either use random port numbers or masquerade to avoid detection), yet it has been utilized here to perform an early classification if any of the P2P-fs applications are still using well-known P2P port numbers for communication.

If σ represents a P2P-fs flow, then similar equation *(3.1)*, this classification process also can be represented as shown in equation *(4.1)* below:

$$C1: f_i = \begin{cases} \sigma & \forall\ p \in d \\ \beta_1 & otherwise \end{cases} \qquad (4.1)$$

where d = database of well-known port numbers used by various P2P-fs applications.

β_1 = non-P2P-fs flow

The traffic flows which are classified as β_1 in equation *(4.1)* are used as an input in the next classification process.

4.3.4 Flow-Level Classification Process (Second Step)

The traffic which remains un-classified as P2P-fs is further fed to the flow-level classification process where initially heuristic-based technique is employed.

4.3.4.1 Heuristic Based Classification

The heuristics that are used in the proposed work are described below:

a) Usage of TCP & UDP (heuristic_1): If both source and destination peers use TCP and UDP protocols simultaneously, then such kind of behavior is shown by P2P applications (such as Skype, BitTorrent, etc.) where TCP could be used for establishing the connection with other peers and UDP could be used for transferring the data amongst those peers (or vice-versa). Hence, such kind of traffic is considered as P2P traffic, but it may not necessarily be P2P-fs traffic. Similar to equation *(3.3)*, this heuristic also can be represented as shown in equation *(4.2)* below:

$$H_1: \quad f_i = \alpha \quad if\ \exists\ (I_{src,i} \in P_{tcp}) \cap (I_{src,i} \in P_{udp}) \qquad (4.2)$$

where α = P2P-flow

b) TCP segment-length or UDP datagram-length ratio (heuristic_2): This heuristic is based on the observation that P2P-fs traffic has a unique packet size distribution in which the TCP segment-length (or UDP datagram-length) for most of the packets in a flow have range either between 0-100 bytes or >1000 bytes. The reason behind such behavior could be that smaller segments/datagrams (i.e., 0-100 bytes in size) are used for data requests, and larger segments/datagrams (i.e.,> 1000 bytes in size) are used for data response. Hence, we define the packet-size-distribution ratio (psd_ratio) of a flow as mentioned in equation *(4.3)*.

$$r = psd\_ratio = \frac{S1 + S2}{t} \quad (4.3)$$

where t = total number of packets of a flow f_i

$S1$ = all those packets of a flow f_i which have segment-length (or datagram-length) in between the range 0-100 bytes.

$S2$ = all those packets of a flow f_i which have segment-length (or datagram-length) greater than 1000 bytes.

Hence, the TCP/UDP flows which have the psd_ratio > threshold-value are classified as P2P-fs traffic. By running P2P-fs applications in controlled environment, it has been observed experimentally that in a traffic flow, at least 70% of the packets (of source & destination) communicate in the of range *S1* and *S2*. Hence, for experimental purposes, the threshold value taken here is 0.70. This heuristic can be represented as shown in equation *(4.4)* below:

$$H_2: \; f_i = \sigma \quad if \; \exists \, r > c; \quad c=0.70 \quad (4.4)$$

c) Usage of ephemeral port numbers (heuristic_3): This heuristic is based on the observation that if both source & destination peers of a flow use ephemeral port numbers (i.e., above 1023) for communication, then such flow is considered as belonging to P2P traffic; but it may not necessarily belong to P2P-fs traffic, since similar behavior could be found in non-file-sharing P2P traffic as well (e.g., VoIP traffic). Similar to equation *(3.2)*, this heuristic also can be represented as shown in equation *(4.5)* below:

$$H_3: \; f_i = \alpha \quad if \; \exists \, (S_{port,i} \in L_e) \cap (D_{port,i} \in L_e) \quad (4.5)$$

where α = P2P-flow

d) Data transfer between peers (heuristic_4): This heuristic is based on the observation that both source & destination peers involved in P2P file-sharing transfer a large number of bytes to each other since the source peer downloads the required data possessed by the destination peer and uploads the data requested by the destination peer. Such kind of traffic differs from FTP traffic in a way that P2P-fs traffic involves the transfer of data in both directions, but FTP traffic involves the transfer of data in a single direction only. Hence, in a flow, if the number of bytes transferred in both directions is greater than a threshold value, then the flow is considered as P2P-fs traffic. By running P2P-fs applications in controlled environment, it has been observed experimentally that in a traffic flow, data transferred in both directions is greater than 3MB (for first 50000 packets transferred on

each side), in comparison to non-P2P-fs applications. Therefore, for experimental purposes, the threshold value taken here is 3MB. Using equation *(3.8)*, this heuristic can also be represented as shown in equation *(4.6)*.

$$H_4:\ f_i = \sigma \quad if\ \exists\, n_s, n_d \mid (t_s >= T) \cap (t_d >= T); \qquad T>=3 \tag{4.6}$$

where n_s = first 50000 packets transferred from source to destination

n_d = first 50000 packets transferred from destination to source

Generally, a heuristic process can be represented as shown in equation *(4.7)* below:

$$H_i = \begin{cases} True & if\ \exists\ f_i \in (\alpha, \sigma); \qquad 1<=i<=4 \\ False & otherwise \end{cases} \tag{4.7}$$

So overall, the packet heuristic process can be represented as shown in equation *(4.8)* below:

$$C2:\ f_i = \begin{cases} \sigma & (H_1 \cap H_2) \cup ((H_3 \cup H_4) \cap H_2) = True \\ \beta_2 & otherwise \end{cases} \tag{4.8}$$

where β_2 = non-P2P-fs flow

The traffic flows which are classified as β_2 in equation *(4.8)* are used as an input in the next classification process. The heuristic-based classification technique is mentioned in Algorithm-4.1 (shown in Table 4.2), where the proposed heuristic rules have been utilized for classifying the traffic either as P2P-fs or non-P2P-fs. Here, we analyze that if source & destination IPs are involved in TCP & UDP communication with each other (i.e., heuristic_1) and their psd_ratio > 0.70 (i.e., heuristic_2), then the flow is labeled as P2P-fs traffic. Otherwise, if source & destination using either the ephemeral ports for communication with each other (i.e., heuristic_3) or transfer data > 3MB to each other (i.e., heuristic_4) and their psd_ratio > 0.70 (i.e., heuristic_2), then such flow is also labeled as P2P-fs traffic. It is to be noted that the heuristics, namely heuristic_1, heuristic_3, and heuristic_4 alone, may not be sufficient to verify whether a flow belongs to P2P-fs traffic or not since such behavior could be shown by non-file-sharing P2P traffic as well (e.g., VoIP traffic).

Table 4.2. Algorithm for performing heuristic-based P2P-fs traffic classification.

Algorithm-4.1: Heuristic-based classification technique

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

src → source peer
dst → destination peer
packet_ratio → ratio of all packets with a *datagram-length-range to total number of packets of a flow
*(*datagram-length-range = 0-100 bytes & >1000 bytes)*
data(src_to_dst) → data transferred from source to destination
data(dst_to_src) → data transferred from destination to source
data(threshold) → 3MB
heuristic_1 → both src & dst uses TCP & UDP simultaneously
heuristic_2 → psd_ratio >= 0.7
heuristic_3 → both src & dst use emphemeral ports
heuristic_4 → data(src_to_dst) & data(dst_to_src) > data(threshold)

Begin

```
1)    while( trafficFlows_not_finished )
2)    {
3)        flow = fetch_next_trafficFlow()
4)        if(flow.heuristic_1() == true)
5)        {
6)            if(flow.heuristic_2() == true)
7)            {
8)                write: flow → P2P_flowTable        //flow classified as P2P-fs
9)            }
10)       }
11)       else if( (flow.heuristic_3() == true)  or  (flow.heuristic_4() == true) )
12)       {
13)           if(flow.heuristic_2() == true)
14)           {
15)               write: flow → P2P_flowTable        //flow classified as P2P-fs
16)           }
17)       }
18)   }
```

End

The flows which remain un-classified as belonging to P2P-fs traffic are fed to the statistical-based classification process where ML algorithm C4.5 decision tree is utilized on the statistical properties of the traffic flow to verify whether remaining traffic contains any trace of P2P-fs traffic or not.

4.3.4.2 Statistical Based Classification

The traffic-flows, which remain unclassified as P2P-fs (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 ML algorithm, namely the C4.5 decision tree to classify the remaining traffic. This process involves the training phase as well as the classification phase. This classification process is similar to the one that is used in section 3.3.4.2; with the only difference being that the ML algorithm here uses training dataset which contains both P2P-fs & non-P2P-fs traffic flow in the training phase. In the classification phase, statistical features of a traffic flow are extracted and fed into the classifier model. If the characteristics of a flow match the distinct characteristics of P2P-fs traffic, then the flow is considered as P2P-fs. Similar to equation *(3.9)*, this classification process is also represented as shown in equation *(4.9)* below:

$$\text{C3: } Gain(S) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Entropy(S_i) \qquad (4.9)$$

The traffic flow features used in this classification process are the same that are mentioned in section 3.3.4.2. They are given as input to the ML algorithm to build a statistical-based classifier for performing the classification. Using equations *(4.1), (4.8)* & *(4.9)*, overall classification of the model can be represented as shown in equation *(4.10)* below:

$$C = C1 \cup C2 \cup C3 \qquad (4.10)$$

4.4 Verification

The implementation process is accomplished using Java programming language along with a java API named jNetPcap [111], which is used to read the network packets for extracting various statistical features such as packet length, port number, IP-address, number of bytes sent or received, etc. Initially, various P2P & non-P2P applications are executed to generate network traffic. As shown in Figure 4.1 (in the previous section), the network packets are then captured from a terminal node of a network with the help of a packet-capturing tool called Wireshark

[16] to create a dataset, which is in the form ".pcap" trace file. Packets are read from the trace file using the jNetPcap library and fed into the classification model.

Initially, the packet-level classification process is utilized where P2P-port-based classification is performed by extracting TCP/UDP port number from each packet header and mapped with the database of well-known P2P port numbers, which various P2P-fs applications may use while communicating. The traffic which remains unclassified as P2P-fs undergoes further analysis in the flow-level classification process, where traffic is analyzed & classified using the heuristic-based technique (as mentioned in Algorithm-4.1 in Table 4.2). Finally, the remaining unclassified traffic undergoes analysis using the statistical-based technique, where ML algorithm C4.5 decision tree is utilized for classification purpose. Here, the ML algorithm is trained using various statistical properties of traffic flow (as mentioned in the previous section). It generates a classification model which is then used to classify the traffic either as P2P-fs or non-P2P-fs. For performing statistical classification with the C4.5 decision tree, an open-source library known as Weka [112] is utilized, which contains the collection of various ML algorithms.

4.4.1 Datasets, Validation, and Experimental Results

The proposed technique classifies the traffic either as P2P-fs traffic or non-P2P-fs traffic (i.e., any traffic other than P2P-fs traffic). It is implemented in java using the jNetPcap library [111] and weka [112]. The metrics which are used to evaluate the proposed methodology are accuracy, false-positive, and false-negative, where accuracy is defined in equation *(4.11)*.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (4.11)$$

Here, TP, TN, FP, and FN refer to true positive, true negative, false positive, and false negative, respectively. Accuracy measures the capability of the classifier in identifying positive and negative cases. The experiment using the proposed technique has been conducted on offline traffic traces, where two datasets have been utilized consisting of both P2P & non-P2P traffic flows, whose details are shown in Table 4.3.

Table 4.3. The number of flows in Dataset-1 and Dataset-2.

Dataset	P2P_fs flows	non-P2P_fs flows	Total
Dataset-1	7599	71399	78998
Dataset-2	20821	10967	31788

The Dataset-1 is UNIBS traffic traces that belongs to the University of Brescia and is available publicly [32] [116]. The Dataset-2 consists of real-traffic traces which have been captured in campus area network in a controlled environment using Wireshark [16] tool, due to which it is known in advance regarding the flows that belong specifically to P2P-fs traffic, and hence such traffic flows are labeled with actual applications accordingly for ground-truth verification. It consists of P2P traffic traces of popular file-sharing applications, namely uTorrent, eMule, Transmission, and qBitTorrent, for analysis purposes. Both datasets (i.e., Dataset-1 and Dataset-2) consist of a mixture of various P2P applications (along with P2P file-sharing applications) and non-P2P applications such as Skype, BitTorrent, Transmission, FTP, HTTP, SSL, DNS, etc.

The experiment conducted on Dataset-2 shows that the packet-level classification process can achieve the classification accuracy of 61.90% only (as shown in Figure 4.3). In addition to that, it has FP and FN rates of 1.53% and 36.56%, respectively (as shown in Figure 4.4). The packets related to connection establishment (i.e., initial communication) among the peers have also been captured in Dataset-2. Here, we observe that some of the peers used well-known P2P port numbers to establish the connection, thereafter which they use random port numbers for communication. Therefore, some of the traffic gets identified at an early stage. However, if this initial communication is missing (or not captured), then the performance of the packet-level classification process would be much poor.

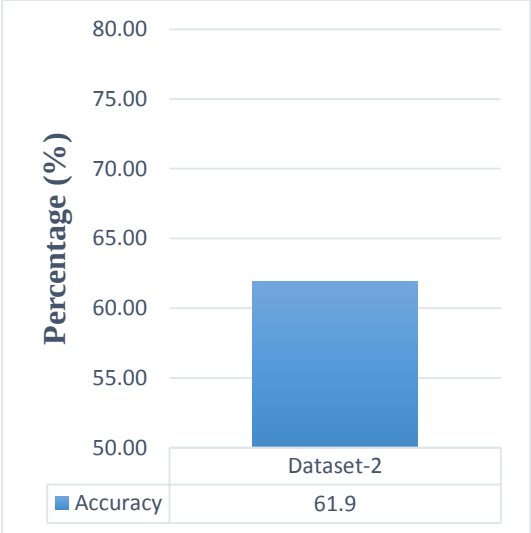


Figure 4.3. Accuracy of the packet-level classification process.

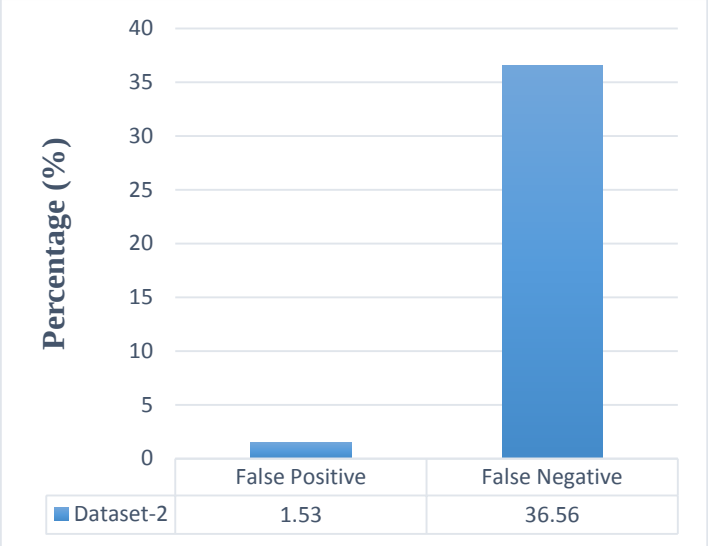


Figure 4.4. FP & FN rates of the packet-level classification process.

The combination of packet-level and flow-level classification process achieved overall accuracy ranging between 98.5% - 99.05% (as shown in Figure 4.5). In addition to that, it has FP and FN rates ranging between 0.9 – 1.2% and 0.1 – 0.2%, respectively (as shown in **Error! Reference source not found.**). The proposed classification technique achieves not only high classification results but also has low overhead (as minimum heuristics are used), classifies both TCP and UDP flows, and works with encrypted traffic as well.

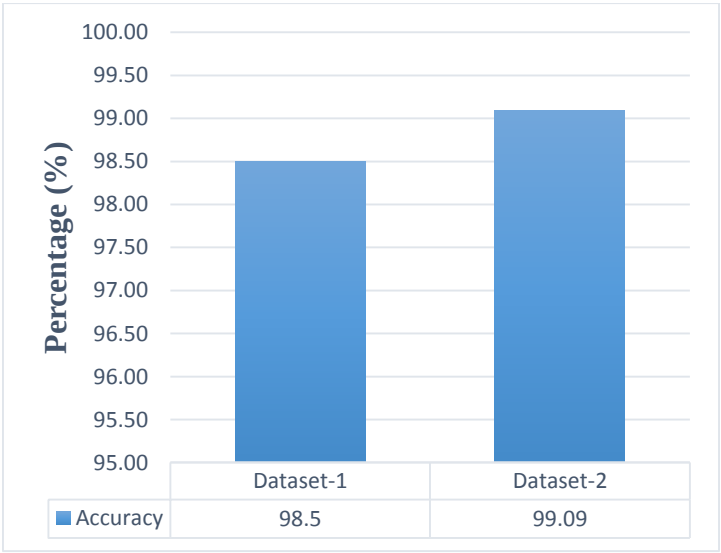


Figure 4.5. Overall classification accuracy of P2P-fs classification technique.

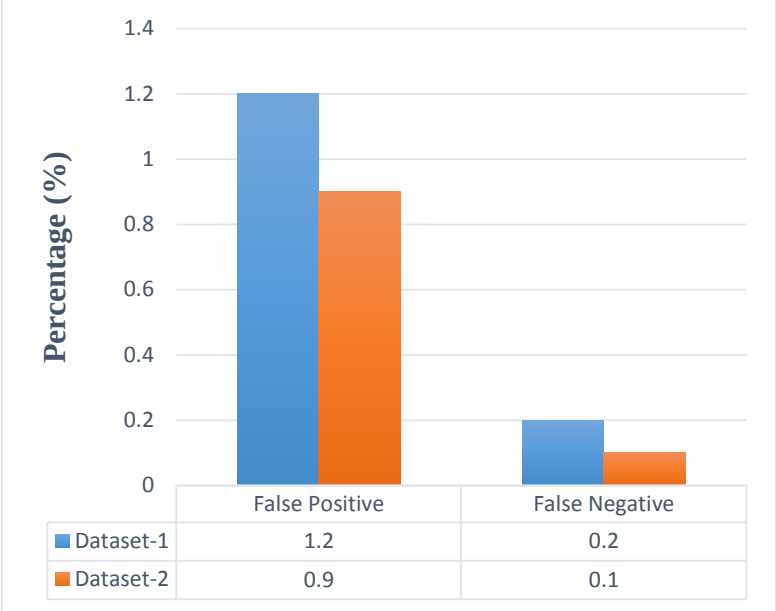


Figure 4.6 FP & FN rates of P2P-fs classification technique.

According to the best of our knowledge, currently, there exists no research work which focuses on classifying various P2P-file-sharing applications from the network traffic. However, there do exist related work in this research area, where authors in [120] proposed their file-sharing classification technique, but their technique specifically focuses on classifying a particular P2P-file-sharing application (i.e. BitTorrent) only, instead of classifying various P2P-file-sharing applications from the network traffic and also they did not specify any information about its applicability on encrypted traffic and accuracy achieved by it. In other related works, for example, the authors in [61], [85], [103], [102] & [110] consider popular P2P-file-sharing applications (like BitTorrent & eMule) in their dataset while performing traffic classification, but their technique classifies P2P traffic overall, instead of classifying P2P-file-sharing applications specifically from the network traffic. Due to this reason, we believe that comparing the classification accuracy of our proposed technique with such studies is unreasonable. Table 4.4 shows the comparative analysis of existing P2P-file-sharing classification techniques with our proposed technique. It specifies the techniques (i.e. port-based, signature-based, statistical-based, machine learning, heuristic-based) utilized by the authors in their approach for classifying the traffic and the ML algorithm used for this purpose. Further, it also specifies their applicability on other aspects, for example, whether they classify TCP/UDP/encrypted traffic and classify specific/generic P2P-fs applications.

Table 4.4. Comparison of the proposed technique with existing P2P-fs classification techniques that specifies the classification technique used/applicability which includes: port (Prt), signature (Sig), statistical (Sta), machine learning (Mch), heuristic/behavior (Heu/Beh), ML-algorithm (Algo), specific/generic P2P-fs classification (Sp/Gn), classify-tcp (TCP), classify-udp (UDP), encryption (Enc).

Ref	Studies	Technique					Algo	Classify		Sp/ Gn	TCP	UDP	Enc
		Prt	Sig	Sta	Mch	Heu/ Beh		P2P-fs	P2P				
[120]	Park et al. (2015)					✓		✓		Sp	✓	✓	--
[61]	Reddy et al. (2015)					✓			✓		✓	✓	--
[85]	Ye and Cho (2014)		✓		✓	✓	REPT ree		✓		✓		✓
[110]	Sajeev and Nair (2016)		✓		✓		C4.5 decisi on tree		✓		✓	✓	--
[102]	Abdalla et al. (2017)			✓	✓				✓		✓	––	--
[103]	Jamil et al. (2019)		✓		✓		ANN, C4.5 decisi on tree		✓		✓	––	--
	Proposed hybrid technique	✓		✓	✓	✓	C4.5 decisi on tree	✓		Gn	✓	✓	✓

4.5 Summary

With the rapid evolution of the internet, various P2P applications and services have emerged and are being adopted by a large number of users. In this chapter, a 2-step P2P traffic classification technique has been proposed, which specifically classifies P2P file-sharing traffic. We specifically focus on P2P file-sharing traffic since it is the largest contributor to P2P internet traffic as a whole. The experimental results show that the proposed technique achieves high accuracy over 98.5%, in classifying P2P file-sharing traffic. In addition to that, the proposed technique has the following capabilities:

a) It is able to classify traffic which either uses TCP or UDP (or both) protocols for communication.
b) It does not use too many heuristics (but just 4) for traffic classification and hence incurs less overhead in comparison to payload-based technique (as discussed in section 2.3 & 2.4).

c) It is able to classify the traffic even if it is encrypted and can be used for real-time classification. This is because the proposed technique does not rely on payload-based technique which is not suitable to classify traffic in high-speed networks or if traffic is encrypted (as discussed in section 2.3 & 2.4).

CHAPTER 5

CLASSIFICATION OF NETWORK TRAFFIC GENERATED BY P2P WEB-SERVICES INCORPORATING VIDEO-STREAMING

5.1 Introduction

Voice over Internet Protocol (VoIP) is an internet technology that provides the ability to transfer voice and media sessions over the IP networks. In recent years, P2P-VoIP applications have become very popular among individuals and enterprises due to high bandwidth connectivity, and low cost in comparison to traditional Public Switched Telephone Network (PSTN) [4]. With the evolution of such applications, it provides better quality of voice & video, free communication between users, and can circumvent the restrictive network environments such as Network Address Translation (NAT) and firewalls.

A VoIP infrastructure typically consists of VoIP clients and signaling servers for call establishment, authentication, and associated services. In addition to that, it may also include additional servers to expedite media transport, achieve traversal of the media path, and interface with PSTN and mobile networks. Recently, there is tremendous growth in VoIP traffic as it is becoming a major communication service for individuals and enterprises [122]. Classifying VoIP traffic can help ISPs and enterprises to prioritize such type of traffic in the network and can enforce policies for network monitoring, load balancing, flow control, managing network bandwidth, providing quality of service, enforcing intrusion detection & prevention services, and auditing.

There are several challenges in classifying VoIP traffic accurately since many applications such as Skype, Google-meet, etc., obfuscate/hide their traffic by making use of random port numbers, encryption, or proprietary protocol for communication. Conventional techniques include port-based & payload-based techniques for classifying network traffic. The port-based technique is the oldest & simplest technique to classify the traffic by using well-known port numbers ranging between 0-1023, which are assigned by IANA [39] to various protocols such as FTP, DNS, HTTP, etc. But, this technique is ineffective in classifying the traffic, which uses random or dynamic port numbers for communication. The payload-based technique (also known as DPI) relies on packet payload and is the most accurate technique in classifying the traffic. It examines the packet payload to search for application-specific signatures and maps it

with the database containing the signatures of previously stored application protocols. However, this technique also suffers from various limitations such as a) unable to deal with encrypted traffic, b) involves a lot of processing load and complexity, c) infeasible in high-speed networks, d) need to find application signatures every time as new application protocol emerges, e) leads to breach of some organization privacy policies by direct inspection of the packet payload, etc. [18] [50] [76]. Therefore, conventional classification techniques, i.e., port-based and payload-based techniques, are ineffective in classifying VoIP traffic, and since they are conventional techniques, so its related work is referred to in [98]. Currently, modern techniques (known as Classification in the Dark) are being employed to classify traffic which makes use of statistical/heuristic-based techniques. Statistical-based techniques classify traffic using statistical features calculated from the traffic, such as packet-length, flow duration, number of packets sent, number of packets received, inter-arrival time of packets, etc. [18]. In contrast, the heuristic-based technique classifies traffic using a pre-defined set of rules by observing the behavioral patterns of the traffic, such as the number of outgoing connections of a host, host acting as both client & server, number of ports used by a host, etc.

The main purpose of this chapter is to classify P2P-VoIP (video) traffic. Many modern VoIP applications have the functionality to make voice calls, video calls, file transfer, and chat, but we specifically focus on classifying video traffic which is generally used for video conferencing or conducting online meetings. For this purpose, we propose a 2-step hybrid classification approach which is categorized into packet-level and flow-level classification processes. The packet-level classification process uses P2P-port based classification technique, whereas the flow-level classification process uses a combination of heuristic-based and statistical-based techniques for classifying VoIP (video) traffic. The experiments have been conducted on three popular VoIP applications, namely Skype, Zoom & Google-meet, and the results show that the proposed technique not only attains high classification accuracy (i.e., 98.6%) but also works with both TCP & UDP protocols and is not affected even if traffic is encrypted.

The remaining chapter is organized as follows. Section 5.2 discusses the related work. Section 5.3 discusses the proposed methodology to classify VoIP (video) traffic. Section 5.4 discusses evaluation criteria and experimental results. Finally, Section 5.5 presents the summary.

5.2 Related Work

Jiang et al. [123] analyzed the network structure of Skype and conducted experiments to observe its new communication pattern after its acquisition by Microsoft. They designed a methodology to detect Skype users in real-time by analyzing the log-in & log-out phases of Skype in each traffic flow. The authors evaluated their technique using 11 hosts (where eight hosts were running Skype) in the actual network environment and claimed that Skype users could be accurately & quickly identified using this approach.

Yuan et al. [124] used an automated packet-sequence signature construction system to construct packet sequence signatures from the application payloads and discovered the sequence of signatures generated by Skype UDP flows. Their technique utilized the combination of login signal detection (to search for '0x02' string present in packet payloads during Skype login session) and destination IP-address lookup (which is one of the destination IP-address used for authentication purpose from the list of IP-addresses used by Skype servers) to identify Skype traffic. The experimental results achieved 98.93% precision and 99.54% recall, but their approach relied on the payload-based technique, which has various limitations.

Lee et al. [125] classified Skype traffic by combining pattern-based and signature-based techniques. It consists of 3 modules which were applied to the traffic in the following sequence: a) login detection (which used pattern-based recognition), b) list-based detection (which used list of IP-port information of Skype detected clients that are fetched during the login detection process), and c) signature-based detection (which used IP correlation and IP-based recognition). The experimental results achieved a detection rate of 95%.

Saqib et al. [126] presented a hybrid technique based on behavioral and statistical analysis to detect and classify VoIP voice packets over IP networks. The 1st step uses behavioral analysis to separate voice and non-voice packets. The 2nd step employed a proposed voice detection algorithm that uses statistical traffic features to further confirm and classify VoIP traffic. But, the proposed technique focused on classifying VoIP voice traffic, and experimental results achieved true positive rates of 93.6% & 95% for offline & online traffic traces, respectively.

Munir et al. [127] performed an analysis on VoIP and non-VoIP traffic and proposed a statistical-based technique that can classify unencrypted, encrypted, and tunneled VoIP-voice traffic. For this purpose, they formulated rules based on threshold values of 9 statistical parameters (i.e., packet rate, mean packet size, the standard deviation of the time difference, etc.) of traffic flows and achieved a detection rate of 97.165%.

Datta et al. [94] proposed a technique to classify Google Hangouts traffic by observing the application behavior. They extracted a set of 7 statistical features by analyzing the connection behavior of Google Hangouts and used them in 3 different ML algorithms (i.e., Naïve Base, AdaBoost & J48) to assess the classification performance. The authors performed dataset collection and evaluation in a controlled network environment. The experimental results found that J48 performs comparatively better in classification and achieved recall rates ranging between 99.99% - 100%.

The technique employed in this chapter focuses on classifying P2P-VoIP (video) traffic specifically since various government organizations and enterprises are currently employing video conferencing to run their businesses [122], and hence such traffic is contributing a lot to the overall P2P traffic on the internet. For this purpose, a combination of heuristic-based and statistical-based techniques has been utilized to classify VoIP traffic.

5.3 P2P-VoIP Traffic Classification Technique

Based on the previous analysis, a multi-level P2P-VoIP traffic classification technique is proposed. It is split 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.

5.3.1 System Model Assumptions

The proposed system model makes similar assumptions as mentioned in section 4.3.1 and are summarized below:

1) All packets of the dataset which does not have IP header are considered insignificant.
2) Small traffic flows where data communication is less than 100 bytes are considered insignificant.
3) The proposed model assumes that P2P-VoIP applications generate only video traffic.

5.3.2 System Model for Classifying P2P-VoIP Traffic

A 2-step classification process (i.e., packet-level & flow-level) has been employed to classify VoIP traffic, as shown in Figure 5.1.

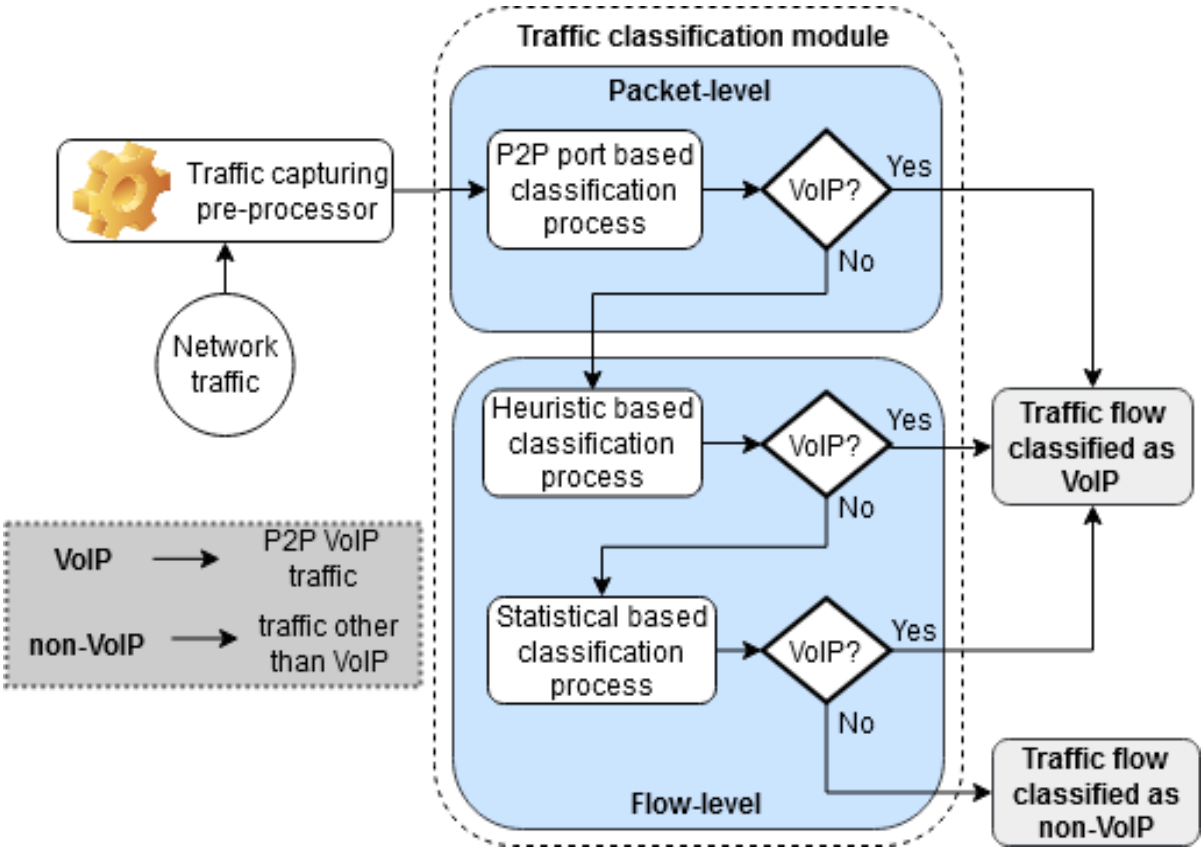
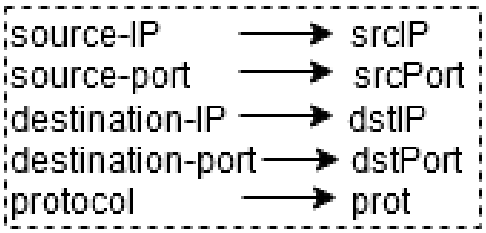


Figure 5.1. P2P-VoIP traffic classification technique.

The packet-level process utilizes a P2P-port-based technique to classify VoIP traffic flows, whereas the flow-level process utilizes the combination of heuristic-based & statistical-based techniques to classify the remaining traffic flows, which could not be classified as VoIP in the 1st step.




```
if (srcPort > dstPort) then
   HashKey (packet) = "srcIP + srcPort + dstIP + dstPort + prot"
else
   HashKey (packet) = "dstIP + dstPort + srcIP + srcPort + prot"
```

Figure 5.2. Calculation of hash-key of a packet.

A traffic flow is generally defined as the combination of 5-tuples (i.e., source-IP, source-port, destination-IP, destination-port, protocol). When two hosts are involved in VoIP communication, packets travel in both directions (i.e., from source to destination and vice-versa). Therefore, during the classification process, the hash-key of packets is calculated by concatenating 5-tuple flow information (as shown in Figure 5.2) so that packets associated with the same traffic flow (traveling in either direction) can be identified; since they will have the similar hash-key. This hash-key is mainly used to find whether the flow of a packet is already classified as VoIP or not. Here, a P2P flow table is used to store both the flow details & packet-hash information of those flows which have already been classified as VoIP.

5.3.3 Packet-Level Classification Process (First Step)

As shown in Figure 5.1, 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.

5.3.3.1 P2P-Port Based Classification

In the packet-level classification process, P2P-port based technique is employed to check if a traffic flow is using VoIP default-port numbers for communication [128] [129] [130] as shown in Table 5.1.

Table 5.1. Default port numbers used by various VoIP applications.

VoIP application	**Default ports used**	
	TCP	**UDP**
Google-meet	443	19302 - 19309
Zoom	443, 80, 8801, 8802	3478, 3479, 8801, 8802
Skype	443	3478 - 3481

For this purpose, the transport-layer port number is extracted from the packet header and mapped with the default port numbers used by various VoIP applications. If a match is found, then the corresponding flow (with which the packet is associated) is classified as VoIP, and flow details are added in the P2P flow table. However, these applications don't need to use default port numbers for communication (e.g., Skype may use other UDP ports in the range 50000 - 60000 also). Various VoIP applications mostly employ random port numbers (or masquerade well-known port numbers such as 80, 443, etc.) for communication, thus making port-based technique ineffective for classification; however, this technique is still employed

here so that if any VoIP application uses default port number for communication, then its flow can be classified at an early stage. In the packet-level classification process, only default UDP port numbers (used by various VoIP applications) have been considered for classification purposes to avoid false-positive cases. If ω represents a P2P-fs flow, then similar equation *(4.1),* this classification process also can be represented as shown in equation *(5.1)* below:

$$C1: f_i = \begin{cases} \omega & \forall\ p \in d \\ \beta_1 & otherwise \end{cases} \qquad (5.1)$$

where d = database of well-known UDP port numbers used by various VoIP applications.

β_1 = non-VoIP flow

The traffic flows which are classified as β_1 in equation *(5.1)* are used as an input in the next classification process.

5.3.4 Flow-Level Classification Process (Second Step)

The traffic which remains un-classified as VoIP in the 1st step undergoes further analysis in the flow-level classification process, where initially heuristic-based technique is employed for classification.

5.3.4.1 Heuristic Based Classification

By analyzing the behavior of various VoIP applications, a set of heuristic rules have been proposed for classifying VoIP (video) traffic, which is described below:

a) Usage of TCP & UDP (heuristic_1): It has been observed that some P2P-VoIP applications (such as Skype) utilize both TCP & UDP protocols simultaneously for communication; where UDP could be used for transferring the data between the peers and TCP could be used for establishing/maintaining the connection between them [61] [113]. Therefore, if source-IP simultaneously uses both TCP & UDP for communication with the destination-IP, then such traffic flow can be considered as VoIP. Similar to equation *(4.2),* this heuristic also can be represented as shown in equation *(5.2)* below:

$$H_1: \ f_i = \alpha \quad if\ \exists\ (I_{src,i} \in P_{tcp}) \cap (I_{src,i} \in P_{udp}) \qquad (5.2)$$

where α = P2P-flow

b) UDP datagram-length ratio (heuristic_2): It has been observed that VoIP applications generally make use of UDP protocol for video communication amongst the peers, and the packet size (i.e., UDP datagram-length) distribution of the majority of the packets lie in between the range: 23-289 bytes & 1037-1222 bytes. Hence, we define the packet-size-distribution ratio (psd_ratio) of a flow as mentioned in equation *(5.3)*.

$$r = psd\_ratio = \frac{S1 + S2}{t} \qquad (5.3)$$

where t = total number of packets of a flow f_i

$S1$ = all those packets of a flow f_i which have segment-length (or datagram-length) in between the range 23-289 bytes.

$S2$ = all those packets of a flow f_i which have segment-length (or datagram-length) in between the range 1037-1222 bytes.

Hence, the TCP/UDP flows which have the psd_ratio > threshold-value are classified as VoIP traffic. For experimental purposes, the threshold value taken here is 0.75. This heuristic can be represented as shown in equation *(5.4)* below:

$$H_2: \; f_i = \omega \quad if \; \exists \, r > c; \quad c=0.75 \qquad (5.4)$$

c) Usage of ephemeral port numbers (heuristic_3): It has been observed that various P2P-VoIP applications utilize ephemeral port numbers (i.e., above 1023) for communication. Hence, in a traffic flow, if both source-IP & destination-IP use ephemeral port numbers for communication, then it can be considered as VoIP. Similar to equation *(4.5)*, this heuristic also can be represented as shown in equation *(5.5)* below:

$$H_3: \; f_i = \alpha \quad if \; \exists \, (S_{port,i} \in L_e) \cap (D_{port,i} \in L_e) \qquad (5.5)$$

where α = P2P-flow

d) Data transfer between peers (heuristic_4): It has been observed that during VoIP (video) communication between two peers, both source & destination peers transfer a large number of bytes to each other. This is because both source & destination peers upload/download the video data to/from each other simultaneously. Therefore, in a traffic flow, if it is found that the number of bytes (i.e., data) transferred in both directions is greater than the threshold value, then such traffic flow can be considered as VoIP. For experimental purposes, the threshold value considered here is 10MB. Such kind of traffic

differs from FTP traffic in a way that VoIP traffic transfers data in both directions simultaneously, whereas FTP traffic transfers data in a single direction only.

Similar to equation *(4.6)*, this heuristic can also be represented as shown in equation *(5.6)*.

$$H_4: f_i = \omega \quad if\ \exists\ (t_s >= T)\ \cap\ (t_d >= T); \quad T>=10 \tag{5.6}$$

Generally, a heuristic process can be represented as shown in equation *(5.7)* below:

$$H_i = \begin{cases} True & if\ \exists\ f_i \in (\alpha, \omega); \quad 1<=i<=4 \\ False & otherwise \end{cases} \tag{5.7}$$

So overall, the packet heuristic process can be represented as shown in equation *(5.8)* below:

$$C2: f_i = \begin{cases} \omega & (H_1 \cap H_2)\ \cup\ ((H_3 \cup H_4) \cap H_2) = True \\ \beta_2 & otherwise \end{cases} \tag{5.8}$$

where β_2 = non-VoIP flow

It is to be noted that heuristic_1, heuristic_3 & heuristic_4 (discussed above) alone are not sufficient to verify whether a traffic flow belongs to VoIP or not, since similar behavior can be seen in P2P file-sharing applications (such as BitTorrent) as well. The algorithm used to classify VoIP traffic is referred to Table 4.2; since the sequence of steps followed to classify VoIP traffic are same; and only difference is that the characteristics used in Table 4.2 are replaced with the following:

Input: Network traffic packets

Output: Traffic-flows classified as VoIP and non-VoIP

psd_ratio → ratio of all packets with a *datagram-length-range to total number of packets of a flow. (*datagram-length-range = 23-289 bytes & 1037-1222 bytes)

data(threshold) → 10MB

heuristic_2 → psd_ratio >= 0.75

In this research work, heuristic-based classification employs the proposed heuristics-rules to classify the traffic flow either as VoIP or non-VoIP. Here, we analyse that if both source & destination IPs of a flow use TCP & UDP simultaneously for communication (i.e. heuristic_1 == true) and their psd_ratio >= 0.75 (i.e. heuristic_2 == true) then the flow is classified as

VoIP. Otherwise, we analyse that if both source & destination IPs of a flow either use ephemeral port numbers (i.e. heuristic_3 == true) or data transfer between them is greater than threshold-value (i.e. heuristic_4 == true) and their psd_ratio >= 0.75 (i.e. heuristic_2 == true), then the flow is classified as VoIP. The traffic flow that remains un-classified as VoIP undergoes further analysis and is fed to the statistical-based classification process, where the ML algorithm (C4.5 decision tree) is applied to the statistical properties of the traffic flow. Therefore, if any VoIP traffic flow goes undetected in the previous processes, then it gets classified at this stage.

5.3.4.2 Statistical Based Classification

The traffic-flows, which remain unclassified as P2P-VoIP (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 ML algorithm, namely the C4.5 decision tree to classify the remaining traffic. This process involves the training phase as well as the classification phase. This classification process is similar to the one that is used in section 3.3.4.2; with the only difference being that the ML algorithm here uses training dataset which contains both P2P-VoIP & non-VoIP traffic flow in the training phase. In the classification phase, statistical features of a traffic flow are extracted and fed into the classifier model. If the characteristics of a flow match the distinct characteristics of VoIP traffic, then the flow is considered as P2P-VoIP. Similar to equation *(4.9)*, this classification process is also represented as shown in equation *(5.9)* below:

$$\text{C3: } Gain(S) = Entropy(S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} Entropy(S_i) \qquad (5.9)$$

The traffic flow features used in this classification process are the same that are mentioned in section 3.3.4.2. They are given as input to the ML algorithm to build a statistical-based classifier for performing the classification. Using equations *(5.1)*, *(5.8)* & *(5.9)*, overall classification of the model can be represented as shown in equation *(5.10)* below:

$$C = C1 \cup C2 \cup C3 \qquad (5.10)$$

5.4 Verification

The implementation process is accomplished using Java programming language along with a java API named jNetPcap [111], which is used to read the network packets for extracting

various statistical features such as packet length, port number, IP-address, number of bytes sent or received, etc. Initially, various P2P & non-P2P applications are executed to generate network traffic. As shown in Figure 5.1 (in the previous section), the network packets are then captured from a terminal node of a network with the help of a packet-capturing tool called Wireshark [16] to create a dataset, which is in the form ".pcap" trace file. Packets are read from the trace file using the jNetPcap library and fed into the classification model.

Initially, the packet-level classification process is utilized where P2P-port-based classification is performed by extracting TCP/UDP port number from each packet header and mapped with the database of well-known P2P port numbers, which various P2P-VoIP applications may use while video communication. The traffic which remains unclassified as P2P-VoIP undergoes further analysis in the flow-level classification process, where traffic is analyzed & classified using the heuristic-based technique (as mentioned in Algorithm-5.1 in **Error! Reference source not found.**). Finally, the remaining unclassified traffic undergoes analysis using the statistical-based technique, where ML algorithm C4.5 decision tree is utilized for classification purpose. Here, the ML algorithm is trained using various statistical properties of traffic flow (as mentioned in the previous section). It generates a classification model which is then used to classify the traffic either as P2P-VoIP or non-VoIP. For performing statistical classification with the C4.5 decision tree, an open-source library known as Weka [112] is utilized, which contains the collection of various ML algorithms.

5.4.1 Datasets, Validation, and Experimental Results

The proposed technique classifies the traffic flow either as P2P-VoIP or non-VoIP. The technique is implemented in java using the jNetPcap library [111] and weka [112] to validate its classification performance. The metrics used for measuring the classification performance are accuracy, false-positive, and false-negative. The experiments have been conducted using offline traffic traces, where two individual datasets consisting of P2P and non-P2P traffic flows have been employed, as shown in Table 5.2. Here, P2P traffic flows consist of both VoIP and file-sharing traffic.

Table 5.2. The number of flows in Dataset-1 and Dataset-2.

Dataset	P2P flows	non-P2P flows	Total
Dataset-1	7599	71399	78998
Dataset-2	20821	10967	31788

Dataset-1 is a publicly available dataset that belongs to the University of Brescia [32] [116]. It consists of both P2P & non-P2P traffic, where P2P traffic traces consist of file-sharing applications (BitTorrent, eDonkey, etc.) and a VoIP application (Skype). Dataset-2 consists of real-traffic traces which are captured in the campus area network (using Wireshark [16]), which comprises a mixture of P2P (including VoIP) & non-P2P traffic. The data capturing was accomplished in a controlled environment where various popular VoIP applications named Google-meet, Skype & Zoom were executed on individual systems for analyzing their pattern of communication, and all other applications were stopped from being executed during this period. Therefore, it was well known in advance regarding the flows which were associated with VoIP traffic and hence were labeled accordingly for ground-truth verification. In addition to that, another system was also made to generate non-VoIP traffic consisting of P2P file-sharing applications (e.g., BitTorrent) & non-P2P applications (e.g., HTTP, HTTPS, DNS, etc.). Hence, overall Dataset-2 consists of both P2P (VoIP & non-VoIP) and non-P2P applications.

During the experiment, a detailed analysis has been conducted on Dataset-2, which shows that the packet-level classification process can achieve the classification accuracy of 56.47% only (as shown in Figure 5.3) and has FP & FN rates of 1.23% & 42.30%, respectively (as shown in Figure 5.4). This is because only some of the traffic flows used default VoIP port numbers (as shown in Table 5.1) during connection establishment; thereafter, random port numbers were used for communication.

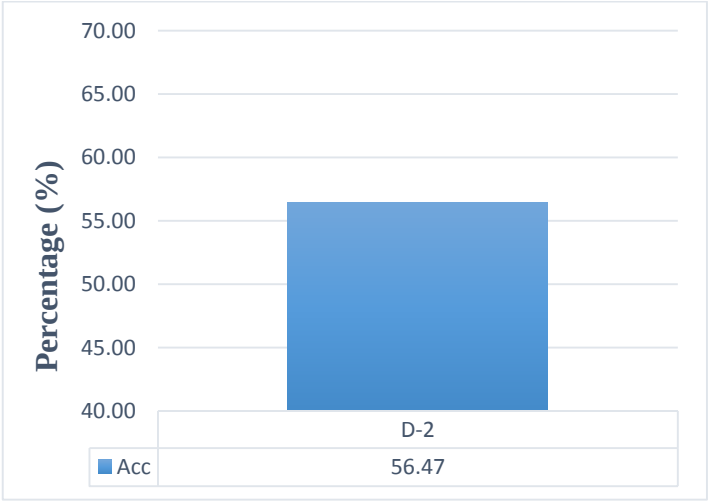


Figure 5.3. Accuracy of the packet-level classification process.

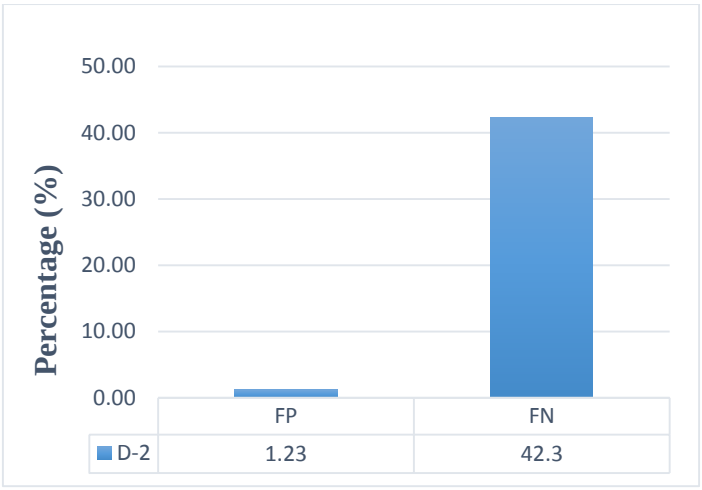


Figure 5.4. FP & FN rates of the packet-level classification process.

It is to be noted that if this initial communication is not captured (or does not contain default VoIP port numbers), then the performance of the packet-level classification process will become very poor. Here, the main purpose of the packet-level classification process is to classify VoIP (video) traffic at an early stage (if some traffic flows are found to be using default VoIP ports) which will reduce the amount of traffic that is required to be analyzed at the flow-level classification process.

The combination of packet-level and flow-level process achieved overall classification accuracy ranging between 98.6% - 99.05% (as shown in Figure 5.5). In addition to that, it has FP & FN rates ranging between 0.1 – 0.2% & 0.85 – 1.2%, respectively (as shown in Figure 5.6). It can be seen that heuristic rules used in the proposed technique work equally with both TCP & UDP traffic and are not affected by encrypted traffic. Hence, the proposed classification technique achieves not only high classification results but also possesses low overhead (since it does not depend upon the DPI technique, which is computationally expensive), classifies both TCP & UDP flows, and works with encrypted traffic as well.

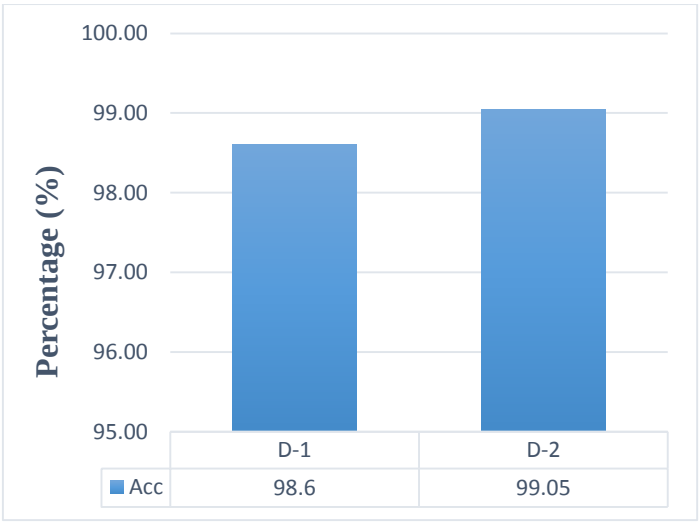


Figure 5.5. Overall classification accuracy of P2P-VoIP classification technique.

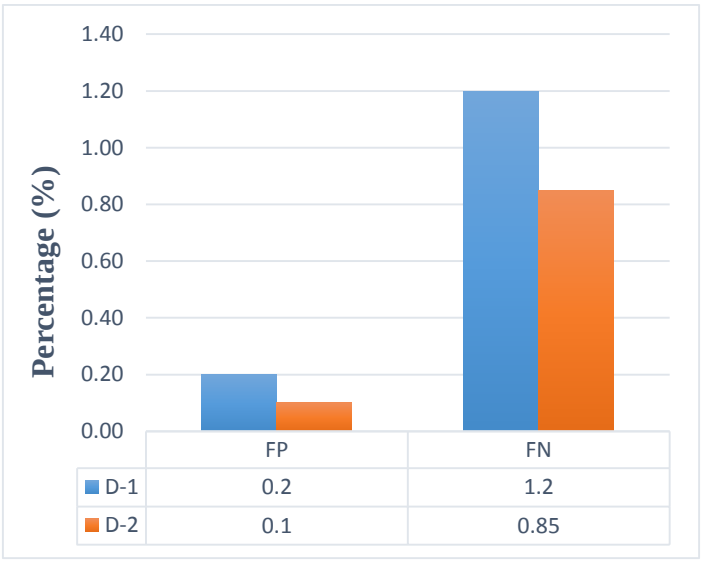


Figure 5.6. FP & FN rates of P2P-VoIP classification technique.

According to the best of our knowledge, currently, there is no research work that focuses on classifying video traffic generated by various P2P-VoIP applications. However, there do exist related work in this research area, where authors in [126] & [127] proposed their VoIP classification technique, but their technique either specifically focus on classifying a particular VoIP application (e.g. Skype) or classifies VoIP-voice traffic only, instead of classifying P2P-VoIP-video traffic from the network traffic. In other related works, for example, the authors in [61] consider popular P2P-VoIP applications (like Skype) in their dataset while performing traffic classification, but their technique classifies P2P traffic overall, instead of classifying P2P-VoIP applications specifically from the network traffic.

Table 5.3. Comparison of the proposed technique with existing VoIP classification techniques that specifies the classification technique used/applicability which includes: port (Port), signature (Sign), statistical (Stat), machine learning (Mach), heuristic/behavior (Heu/Beh), ML-algorithm (Algo), specific/generic classification (Sp/Gn), classify-tcp (TCP), classify-udp (UDP), encryption (Enc), accuracy (Acc).

Ref	Studies	Technique					Algo	Sp/ Gn	TCP	UDP	Enc	Acc (%)
		Port	Sign	Stat	Mach	Heu/ Beh						
[94]	Datta et al. (2015)					✓	J48 decision tree	Sp	✓	✓	--	--
[123]	Jiang et al. (2016)					✓		Sp	✓	✓	✓	--
[124]	Yuan et al. (2014)		✓			✓		Sp	✓	✓	--	--
[125]	Lee et al. (2017)		✓			✓		Sp	✓	✓	--	95%
[126]	Saqib et al. (2017)			✓		✓		Gn	✓	✓	✓	95%
[127]	Munir et al. (2016)			✓		✓		Gn	✓	✓	✓	97.16
	Proposed hybrid technique	✓		✓	✓	✓	C4.5 decision tree	Gn	✓	✓	✓	98.30

Table 5.3 shows the comparative analysis of existing VoIP classification techniques with our proposed technique. It specifies the techniques (i.e. port-based, signature-based, statistical-based, machine learning, heuristic-based) utilized by the authors in their approach for classifying the traffic and the ML algorithm used for this purpose. Further, it also specifies their applicability on other aspects, for example, whether they classify TCP/UDP/encrypted traffic, classify specific/generic VoIP applications, and overall accuracy achieved. There exist various metrics (such as recall, precision, accuracy, etc.) which can be used to evaluate the performance of a traffic classification technique. Various authors used different metrics to evaluate their classification technique. So, in Table 5.3, the accuracy achieved by some studies is not displayed, as the authors used different metrics for the performance evaluation of their technique.

5.5 Summary

P2P-VoIP applications have become prevalent in recent years and are currently being used extensively by various companies, enterprises, and government organizations globally to run their businesses. VoIP traffic which involves video communication consumes a lot of network bandwidth, and hence it needs to be classified so that it can be managed/prioritized by the ISPs

and network administrators to maintain the Quality of Service of various network applications. In this chapter, a 2-step hybrid approach is employed, which combines heuristic-based and statistical-based techniques to classify VoIP (video) traffic. The experimental results display that the proposed technique attains high classification accuracy of over 98.6%. In addition to that, the proposed technique has the following capabilities:

a) It classifies both TCP & UDP traffic flows.
b) It works with encrypted traffic and can be used for real-time classification as well.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

P2P architecture consists of distributed systems interconnected to each other where each participating peer can share its resources such as files, storage capacity, processing power, etc., to the other peers over the network without requiring a centralized server. By utilizing these resources, various P2P services are implemented, such as file-sharing, audio/video streaming, online gaming, etc., which are accessible by all the peers of a network. Since the past decade, such services are widely used and accessed via specific P2P applications that bring many conveniences such as reliability, easy & quick file-sharing, high performance, reduced cost, etc. However, such applications pose various challenges to the ISPs & enterprises, such as providing excellent broadband experience to customers, purchasing costly backbone links & up-streaming bandwidth, etc. Considering the overall network traffic, which is composed of traffic from various application protocols (SMTP, FTP, DNS, HTTP, P2P, HTTPS, etc.), traffic from P2P applications alone consumes a significant portion of the available 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 to perform various network-related tasks such as:

- Network bandwidth planning
- Policy-based traffic management
- Fault diagnosis
- QoS analysis for applications
- Accurate accounting for billing
- Lawful interception for security-related issues

Conventional techniques for traffic classification, such as port-based & payload-based, are ineffective in classifying P2P traffic due to various limitations associated with them. Therefore, a modern approach known as Classification in the Dark is currently adopted to classify P2P traffic with high accuracy. It allows ISPs or network administrators to either prioritize, limit or

completely ban P2P traffic for maintaining Quality of Service for various applications in their network.

Following are the contribution of this thesis:

Initially, this research work focuses on classifying P2P traffic (overall) in the network. For this purpose, a multi-level P2P traffic classification technique is adopted, which is sub-divided into the packet-level and flow-level classification processes. Firstly, the P2P-port-based classification technique is employed for P2P traffic classification. Here, the TCP/UDP port number is extracted from the packet header and mapped with the database of well-known P2P port numbers which various P2P applications may use. If the traffic remains unclassified as P2P, then a set of proposed heuristic rules (i.e., packet-level heuristics & flow-level heuristics) is employed to perform P2P traffic classification. If the traffic remains unclassified as P2P, it undergoes further analysis using the statistical-based technique, which utilizes ML algorithm C4.5 decision tree on the traffic flow's statistical properties to classify the traffic either P2P or non-P2P.

Further, we focus on classifying network traffic generated by various P2P file-download applications such as uTorrent, eMule, etc. For this purpose, a 2-step traffic classification technique is adopted, which specifically classifies P2P file-sharing traffic. Here, we specifically focus on P2P file-sharing traffic since it is the most significant contributor to P2P internet traffic as a whole. The classification process is categorized into two levels, namely packet-level & flow-level classification. In the packet-level classification process, the P2P-port-based technique is initially employed for classification by extracting the port number from the packet header and mapped with well-known P2P port numbers, which various P2P-fs applications may use. The traffic which remains un-classified as P2P-fs is fed to the flow-level classification process, which consists of heuristic-based & statistical-based techniques. The heuristic-based technique uses proposed heuristic rules to classify the traffic either as P2P-fs or non-P2P-fs. The traffic flows that remain un-classified as P2P-fs traffic are further fed to the statistical-based classification technique that utilizes ML algorithm C4.5 decision tree on the traffic flow's statistical properties to verify whether the remaining traffic contains any trace of P2P-fs traffic or not.

Further, we focus on classifying network traffic generated by various P2P-VoIP applications such as Skype, Google-meet, etc. Many modern VoIP applications possess the functionality to make voice calls, video calls, file transfers, and chat. But, we specifically focus

on classifying video traffic which is generally used for video conferencing or conducting online meetings. We employed a 2-step hybrid classification approach that is categorized into packet-level and flow-level classification processes. In the packet-level classification process, initially, P2P-port based technique is employed for classification by extracting the port number from the packet header and mapped with well-known P2P port numbers, which various P2P-VoIP applications may use. The traffic which remains un-classified as P2P-VoIP is then fed to the flow-level classification process, which consists of heuristic-based & statistical-based techniques. The heuristic-based technique uses proposed heuristic rules to classify the traffic either as VoIP or non-VoIP. The traffic flows that remain un-classified as VoIP traffic are further fed to the statistical-based classification technique that utilizes ML algorithm C4.5 decision tree on the traffic flow's statistical properties to verify whether the remaining traffic contains any trace of VoIP traffic or not.

The proposed 2-step hybrid techniques used in this research work for classifying P2P traffic (as a whole or specifically) not only achieve high classification accuracy but also possess the following capabilities:

- They work on both TCP & UDP protocols.
- They use minimum heuristics for classification (and hence have less overhead).
- They work with encrypted traffic and can be used for real-time classification as well.

However, there are certain limitations in the proposed techniques, which are mentioned below:

- They 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 various P2P applications can utilize.
- They do not perform a fine-grained classification in identifying traffic of specific P2P applications.

6.1 Future work

Although we proposed a hybrid model to classify P2P traffic with good accuracy, but still there is a scope of improvement. By considering the limitations mentioned in previous section, the proposed classification technique can be enhanced further.

The proposed classification model can produce false-positive cases if any malicious application uses well known P2P default ports (that various P2P applications can utilize) for communication. Here, the classification model can be improved further to avoid such cases.

Furthermore, various P2P applications (e.g. VoIP, BitTorrent, etc.) share similar characteristics while communicating with other peers over the internet. Therefore, there is a need to explore some unique characteristics of various P2P applications (by analyzing the communication pattern of each), so that a single classification model is able to identify each P2P application uniquely. For example, if various P2P applications like BitTorrent, Skype, etc. are flowing through the network, then the classification model should be able to classify the traffic of each application uniquely. This will help ISPs or network administrators to prioritize the traffic of specific P2P applications (e.g. Skype, Google-meet, etc.), by limiting the traffic bandwidth of other P2P applications (e.g. BitTorrent, etc.) that are flowing through the network. Therefore, classification model needs to be enhanced such that it can perform fine-grained P2P traffic classification by identifying the traffic generated by specific P2P applications uniquely.

REFERENCES

[1] J. Hurley, E. Garcia-Palacios and S. Sezer, "Classification of P2P and HTTP using specific protocol characteristics," in *Meeting of the European Network of Universities and Companies in Information and Communication Engineering*, Springer, 2009, pp. 31-- 40.

[2] M. Mohammadi, B. Raahemi, A. Akbari, H. Moeinzadeh and B. Nasersharif, "Genetic-based minimum classification error mapping for accurate identifying Peer-to-Peer applications in the internet traffic," *Expert Systems with applications,* vol. 38, no. 6, pp. 6417--6423, 2011.

[3] S. Sen and J. Wang, "Analyzing peer-to-peer traffic across large networks," *ACM,* pp. 137--150, 2002.

[4] "Global Internet Phenomena," Sandvine, 2019. [Online]. Available: https://www.sandvine.com/phenomena.

[5] N. B. Azzouna and F. Guillemin, "Analysis of ADSL traffic on an IP backbone link," in *GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489),* 2003.

[6] H. Schulze and K. Mochalski, "Internet Study 2008/2009," *Ipoque Report,* vol. 37, pp. 351--362, 2009.

[7] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy and M. Faloutsos, "File-sharing in the Internet: a characterization of P2P traffic in the backbone," *University of California, Riverside, USA, Tech. Rep,* 2003.

[8] A. Madhukar and C. Williamson, "A longitudinal study of P2P traffic classification," in *14th IEEE International Symposium on Modeling, Analysis, and Simulation*, 2006.

[9] C. Williamson, "Internet traffic measurement," *IEEE internet computing,* vol. 5, no. 6, pp. 70--74, 2001.

[10] "Enterprise network monitoring tools – network security system – application performance monitoring," [Online]. Available: http://www.endace.com.

[11] "IPOQUE (2015) Bandwidth management with deep packet inspection," [Online]. Available: http://www.ipoque.com.

[12] "WildPackets: Network analyzer, voip monitoring, protocol analysis," [Online]. Available: http://www.wildpackets.com.

[13] "Intelligent real-time network analysis," [Online]. Available: http://www.napatech.com.

[14] "SNORT," [Online]. Available: http://www.snort.org.

[15] "Bro intrusion detection system," [Online]. Available: http://bro-ids.org.

[16] “Wireshark,” [Online]. Available: https://www.wireshark.org. [Accessed 2019].

[17] “ETTERCAP,” [Online]. Available: http://ettercap.sourceforge.net.

[18] J. V. Gomes, P. R. Inacio, M. Pereira, M. M. Freire and P. P. Monteiro, “Detection and classification of peer-to-peer traffic: A survey,” *ACM Computing Surveys (CSUR),* vol. 45, no. 3, p. 30, 2013.

[19] K. C. Claffy, H.-W. Braun and G. C. Polyzos, “A parameterizable methodology for Internet traffic flow profiling,” *IEEE Journal on selected areas in communications,* vol. 13, no. 8, pp. 1481--1494, 1995.

[20] D. Moore, K. Keys, R. Koga and E. Lagache, “CoralReef software suite as a tool for system and network administrators,” in *Usenix LISA,* 2001.

[21] “CISCO NETFLOW,” [Online]. Available: https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html.

[22] M. Alllman and V. Paxson, “Issues and etiquette concerning use of shared measurement data,” in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007.

[23] “TCPDUMP/LIBPCAP public repository,” [Online]. Available: http://www.tcpdump.org.

[24] “WINDUMP. tcpdump forWindows usingWinPcap,” [Online]. Available: http://www.winpcap.org/windump.

[25] R. E. Jurga and M. M. Hulboj, “Packet sampling for network monitoring,” CERN—HP Procurve openlab project, 2007.

[26] N. Duffield and others, “Sampling for passive internet measurement: A review,” *Statistical Science,* vol. 19, no. 3, pp. 472--498, 2004.

[27] A. Sperotto, R. Sadre, F. Van Vliet and A. Pras, “A labeled data set for flow-based intrusion detection,” in *International Workshop on IP Operations and Management*, Springer, 2009, pp. 39--50.

[28] D. Zuev and A. W. Moore, “Traffic classification using a statistical approach,” in *International workshop on passive and active network measurement*, Springer, 2005, pp. 321--324.

[29] T. Karagiannis, K. Papagiannaki and M. Faloutsos, “BLINC: multilevel traffic classification in the dark,” *ACM SIGCOMM computer communication review,* vol. 35, no. 4, pp. 229--240, 2005.

[30] L. Salgarelli, F. Gringoli and T. Karagiannis, “Comparing traffic classifiers,” *ACM SIGCOMM Computer Communication Review,* vol. 37, no. 3, pp. 65--68, 2007.

[31] M. Canini, W. Li, A. W. Moore and R. Bolla, “GTVS: Boosting the collection of application traffic ground truth,” in *International Workshop on Traffic Monitoring and Analysis*, Springer, 2009, pp. 54--63.

[32] F. Gringoli, L. Salgarelli, M. Dusi, N. Cascarano, F. Risso and others, “Gt: picking up the truth from the ground for internet traffic,” *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 5, pp. 12--18, 2009.

[33] G. Szab'o, D. Orincsay, S. Malomsoky and I. Szab'o, “On the validation of traffic classification algorithms,” in *International conference on passive and active network measurement*, Springer, 2008, pp. 72--81.

[34] J. Makhoul, F. Kubala, R. Schwartz, R. Weischedel and others, “Performance measures for information extraction,” in *Proceedings of DARPA broadcast news workshop*, Herndon, VA, 1999, pp. 249--252.

[35] D. L. Olson and D. Delen, Advanced data mining techniques, Springer Science & Business Media, 2008.

[36] T. T. Nguyen and G. J. Armitage, “A survey of techniques for internet traffic classification using machine learning,” *IEEE Communications Surveys and Tutorials*, vol. 10, no. 1-4, pp. 56--76, 2008.

[37] Y. Wang, Statistical Techniques for Network Security: Modern Statistically-Based Intrusion Detection and Protection: Modern Statistically-Based Intrusion Detection and Protection, Igi Global, 2008.

[38] B. Raahemi, W. Zhong and J. Liu, “Peer-to-peer traffic identification by mining IP layer data streams using concept-adapting very fast decision tree,” in *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, vol. 1, IEEE, 2008, pp. 525--532.

[39] “Service Name and Transport Protocol Port Number Registry,” [Online]. Available: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

[40] A. W. Moore and K. Papagiannaki, “Toward the accurate identification of network applications,” in *International Workshop on Passive and Active Network Measurement*, Springer, 2005, pp. 41--54.

[41] T. Karagiannis, A. Broido, N. Brownlee, K. C. Claffy and M. Faloutsos, “Is p2p dying or just hiding?[p2p traffic measurement],” in *IEEE Global Telecommunications Conference, 2004. GLOBECOM'04.*, 2004.

[42] M. Roughan, S. Sen, O. Spatscheck and N. Duffield, “Class of service mapping for QoS: A statistical signature-based approach to IP traffic classification,” in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004.

[43] S. Sen, O. Spatscheck and D. Wang, “Accurate, scalable in-network identification of p2p traffic using application signatures,” in *Proceedings of the 13th international conference on World Wide Web*, 2004.

[44] T. Karagiannis, A. Broido, M. Faloutsos and others, “Transport layer identification of P2P traffic,” in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, 2004.

[45] K. Wang and S. J. Stolfo, “Anomalous payload-based network intrusion detection,” in *International workshop on recent advances in intrusion detection*, Springer, 2004, pp. 203--222.

[46] T. Song and Z. Zhou, “File-aware P2P traffic classification: An aid to network management,” *Peer-to-Peer Networking and Applications,* vol. 6, no. 3, pp. 325--339, 2013.

[47] W. H. Turkett Jr, A. V. Karode and E. W. Fulp, “In-the-dark network traffic classification using support vector machines,” *AAAI*, pp. 1745--1750, 2008.

[48] E. P. Freire, A. Ziviani and R. M. Salles, “Detecting skype flows in web traffic,” in *NOMS 2008-2008 IEEE Network Operations and Management Symposium*, IEEE, 2008, pp. 89--96.

[49] E. P. Freire, A. Ziviani and R. M. Salles, “Detecting VoIP calls hidden in web traffic,” *IEEE Transactions on Network and Service Management*, vol. 5, no. 4, pp. 204--214, 2008.

[50] J. V. Gomes, P. R. Inacio, M. M. Freire, M. Pereira and P. P. Monteiro, “Analysis of peer-to-peer traffic using a behavioural method based on entropy,” in *2008 IEEE International Performance, Computing and Communications Conference*, 2008.

[51] M.-F. Sun and J.-T. Chen, “Research of the traffic characteristics for the real time online traffic classification,” *The Journal of China Universities of Posts and Telecommunications*, vol. 18, no. 3, pp. 92--98, 2011.

[52] A. Moore, D. Zuev and M. Crogan, “Discriminators for use in flow-based classification,” 2013.

[53] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule and K. Salamatian, “Traffic classification on the fly,” *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 2, pp. 23--26, 2006.

[54] J. He, Y.-x. Yang, Y. Qiao and W.-p. Deng, “Fine-grained P2P traffic classification by simply counting flows,” *Frontiers of Information Technology & Electronic Engineering*, vol. 16, no. 5, pp. 391--403, 2015.

[55] K. Yang, B. Wang and Z. Zhang, “A method of identifying P2P live streaming based on union features,” in *2013 IEEE 4th International Conference on Software Engineering and Service Science*, IEEE, 2013, pp. 426--429.

[56] T. Qin, L. Wang, D. Zhao and M. Zhu, “CUFTI: Methods for core users finding and traffic identification in P2P systems,” *Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 424--435, 2016.

[57] Q. Zhang, Y. Ma, P. Zhang, J. Wang and X. Li, “Netflow Based P2P detection in UDP traffic,” in *Fifth International Conference on Intelligent Control and Information Processing*, 2014.

[58] M. Perenyi, T. D. Dang, A. Gefferth and S. Molnar, “Identification and analysis of peer-to-peer traffic,” *Journal of Communications,* vol. 1, no. 7, pp. 36--46, 2006.

[59] W. John and S. Tafvelin, “Heuristics to classify internet backbone traffic based on connection patterns,” in *2008 International Conference on Information Networking*, IEEE, 2008, pp. 1--5.

[60] W.-m. Hong, “A novel method for P2P traffic identification,” *Procedia Engineering,* vol. 23, no. Elsevier, pp. 204--209, 2011.

[61] J. M. Reddy and C. Hota, “Heuristic-based Real-Time P2P Traffic Identification,” in *2015 International Conference on Emerging Information Technology and Engineering Solutions*, 2015.

[62] A. Bashir, C. Huang, B. Nandy and N. Seddigh, “Classifying P2P activity in Netflow records: A case study on BitTorrent,” in *2013 IEEE International Conference on Communications (ICC)*, IEEE, 2013, pp. 3018--3023.

[63] A. McGregor, M. Hall, P. Lorier and J. Brunskill, “Flow clustering using machine learning techniques,” in *International workshop on passive and active network measurement*, Springer, 2004, pp. 205--214.

[64] A. W. Moore and D. Zuev, “Internet traffic classification using bayesian analysis techniques,” in *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, 2005, pp. 50--60.

[65] P. A. Branch, A. Heyde and G. J. Armitage, “Rapid identification of Skype traffic flows,” in *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*, 2009, pp. 91--96.

[66] S. Schmidt and M. Soysal, “An intrusion detection based approach for the scalable detection of P2P traffic in the national academic network backbone,” in *2006 International Symposium on Computer Networks*, IEEE, 2006, pp. 128--133.

[67] J. Cao, A. Chen, I. Widjaja and N. Zhou, “Online identification of applications using statistical behavior analysis,” in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, IEEE, 2008, pp. 1--6.

[68] D. Angevine and N. Zincir-Heywood, “A preliminary investigation of Skype traffic classification using a minimalist feature set,” in *2008 Third International Conference on Availability, Reliability and Security*, IEEE, 2008, pp. 1075--1079.

[69] Y.-H. Wang, V. Gau, T. Bosaw, J.-N. Hwang, A. Lippman, D. Lieberman and I.-C. Wu, “Generalization performance analysis of flow-based peer-to-peer traffic identification,” in *2008 IEEE Workshop on Machine Learning for Signal Processing*, IEEE, 2008, pp. 267--272.

[70] A. Dainotti, W. De Donato, A. Pescape and P. S. Rossi, "Classification of network traffic via packet-level hidden markov models," in *IEEE GLOBECOM 2008-2008 IEEE Global Telecommunications Conference*, IEEE, 2008, pp. 1--5.

[71] S. Valenti, D. Rossi, M. Meo, M. Mellia and P. Bermolen, "Accurate, fine-grained classification of P2P-TV applications by simply counting packets," in *International Workshop on Traffic Monitoring and Analysis*, Springer, 2009, pp. 84--92.

[72] H. Liu, W. Feng, Y. Huang and X. Li, "A peer-to-peer traffic identification method using machine learning," in *2007 International Conference on Networking, Architecture, and Storage (NAS 2007)*, IEEE, 2007, pp. 155--160.

[73] B. Raahemi, A. Kouznetsov, A. Hayajneh and P. Rabinovitch, "Classification of Peer-to-Peer traffic using incremental neural networks (Fuzzy ARTMAP)," in *2008 Canadian Conference on Electrical and Computer Engineering*, IEEE, 2008, pp. 719--724.

[74] Y. Hu, D.-M. Chiu and J. C. Lui, "Application identification based on network behavioral profiles," in *2008 16th interntional workshop on quality of service*, IEEE, 2008, pp. 219--228.

[75] Y. Hu, D.-M. Chiu and J. C. Lui, "Profiling and identification of P2P traffic," *Computer Networks*, vol. 53, no. 6, pp. 849--863, 2009.

[76] S.-M. Liu and Z.-X. Sun, "Active learning for P2P traffic identification," *Peer-to-Peer Networking and Applications*, vol. 8, no. 5, pp. 733--740, 2015.

[77] D. Jiang and L. Tao, "P2P traffic identification research based on the SVM," in *2013 22nd Wireless and Optical Communication Conference*, IEEE, 2013, pp. 683--686.

[78] J. Gong, W. Wang, P. Wang and Z. Sun, "P2P Traffic Identification Method based on an Improvement Incremental SVM Learning Algorithm," in *2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2014.

[79] S. Deng, J. Luo, Y. Liu, X. Wang and J. Yang, "Ensemble learning model for P2P traffic identification," in *2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2014.

[80] H. Jie, Y. Yuexiang, Q. Yong and T. Chuan, "Accurate classification of P2P traffic by clustering flows," *China Communications*, vol. 10, no. 11, pp. 42--51, 2013.

[81] C. Bozdogan, Y. Gokcen and I. Zincir, "A preliminary investigation on the identification of peer to peer network applications," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, 2015.

[82] I. Dedinski, H. De Meer, L. Han, L. Mathy, D. P. Pezaros, J. S. Sventek and X. Zhan, "Cross-layer peer-to-peer traffic identification and optimization based on active networking," in *IFIP International Working Conference on Active Networks*, Springer, 2005, pp. 13--27.

[83] D. Adami, C. Callegari, S. Giordano, M. Pagano and T. Pepe, “A real-time algorithm for skype traffic detection and classification,” in *Smart Spaces and Next Generation Wired/Wireless Networking*, Springer, 2009, pp. 168--179.

[84] J. Yan, Z. Wu, H. Luo and S. Zhang, “P2P Traffic Identification Based on Host and Flow Behaviour Characteristics,” *Cybernetics and Information Technologies*, vol. 13, no. 3, pp. 64--76, 2013.

[85] W. Ye and K. Cho, “Hybrid P2P traffic classification with heuristic rules and machine learning,” *Soft Computing*, vol. 18, no. 9, pp. 1815--1827, 2014.

[86] W. Ye and K. Cho, “P2P and P2P botnet traffic classification in two stages,” *Soft Computing*, vol. 21, no. 5, pp. 1315--1326, 2017.

[87] D. Wang, L. Zhang, Z. Yuan, Y. Xue and Y. Dong, “Characterizing application behaviors for classifying p2p traffic,” in *2014 International Conference on Computing, Networking and Communications (ICNC)*, 2014.

[88] Z. Yang, L. Li, Q. Ji and Y. Zhu, “Cocktail method for BitTorrent traffic identification in real time.,” *JCP*, vol. 7, no. 1, pp. 85--95, 2012.

[89] M. Korczy'nski and A. Duda, “Markov chain fingerprinting to classify encrypted traffic,” in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, IEEE, 2014, pp. 781--789.

[90] R. Alshammari and A. N. Zincir-Heywood, “Identification of VoIP encrypted traffic using a machine learning approach,” *Journal of King Saud University-Computer and Information Sciences*, vol. 27, no. 1, pp. 77--92, 2015.

[91] Y. Kumano, S. Ata, N. Nakamura, Y. Nakahira and I. Oka, “Towards real-time processing for application identification of encrypted traffic,” in *2014 International Conference on Computing, Networking and Communications (ICNC)*, IEEE, 2014, pp. 136--140.

[92] X. Wang, J. He and Y. Yang, “Identifying P2P network activities on encrypted traffic,” in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, IEEE, 2014, pp. 893--899.

[93] Y. Du and R. Zhang, “Design of a method for encrypted P2P traffic identification using K-means algorithm,” *Telecommunication Systems*, vol. 53, no. 1, pp. 163--168, 2013.

[94] J. Datta, N. Kataria and N. Hubballi, “Network traffic classification in encrypted environment: a case study of google hangout,” in *Twenty First National Conference on Communications (NCC)*, 2015.

[95] L. Dai, J. Yang and L. Lin, “A comprehensive system for P2P classification,” in *2010 2nd IEEE InternationalConference on Network Infrastructure and Digital Content*, 2010.

[96] H. Chu, H. Yi and X. Zhang, “A new P2P traffic identification methodology based on flow statistics,” in *2011 IEEE 3rd International Conference on Communication Software and Networks*, 2011.

[97] R. Keralapura, A. Nucci and C.-N. Chuah, “A novel self-learning architecture for p2p traffic classification in high speed networks,” *Computer Networks,* vol. 54, no. 7, pp. 1055--1068, 2010.

[98] M. Bhatia and M. K. Rai, “Identifying P2P traffic: A survey,” *Peer-to-Peer Networking and Applications,* vol. 10, no. 5, pp. 1182--1203, 2017.

[99] “Controlling P2P Traffic,” 2003. [Online]. Available: https://www.lightreading.com/controlling-p2p-traffic/d/d-id/598203&page_number=2.

[100] C.-M. Tseng, G.-T. Huang and T.-J. Liu, “P2P traffic classification using clustering technology,” in *2016 IEEE/SICE International Symposium on System Integration (SII)*, 2016.

[101] L. Chuan, C. Wang, H. Jixiong and Z. Ye, “Peer to Peer Traffic Identification Using Support Vector Machine and Bat-Inspired Optimization Algorithm,” in *2017 12th International Conference on Computer Science and Education (ICCSE)*, 2017.

[102] B. M. A. Abdalla, H. A. Jamil, M. Hamdan, J. S. Bassi, I. Ismail and M. N. Marsono, “Multi-stage feature selection for on-line flow peer-to-peer traffic identification,” in *Asian Simulation Conference*, 2017.

[103] H. A. Jamil, B. M. Ali, M. Hamdan and A. E. Osman, “Online P2P Internet Traffic Classification and Mitigation Based on Snort and ML,” *European Journal of Engineering Research and Science,* vol. 4, no. 10, pp. 131--137, 2019.

[104] Z. Nazari, M. Noferesti and R. Jalili, “DSCA: an inline and adaptive application identification approach in encrypted network traffic,” in *Proceedings of the 3rd International Conference on Cryptography, Security and Privacy*, 2019.

[105] W. Ye and K. Cho, “Two-step p2p traffic classification with connection heuristics,” in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, 2013.

[106] R. U. Khan, R. Kumar, M. Alazab and X. Zhang, “A Hybrid Technique To Detect Botnets, Based on P2P Traffic Similarity,” in *2019 Cybersecurity and Cyberforensics Conference (CCC)*, 2019.

[107] J. Li, S. Zhang, Y. Lu and J. Yan, “Hybrid Internet traffic classification technique,” *Journal of Electronics (China),* vol. 26, no. 1, pp. 101--112, 2009.

[108] Z. Chen, B. Yang, Y. Chen, A. Abraham, C. Grosan and L. Peng, “Online hybrid traffic classifier for peer-to-peer systems based on network processors,” *Applied Soft Computing*, vol. 9, no. 2, pp. 685--694, 2009.

[109] L. M. Nair and G. Sajeev, "Internet traffic classification by aggregating correlated decision tree classifier," in *2015 Seventh International Conference on Computational Intelligence, Modelling and Simulation (CIMSim)*, 2015.

[110] G. Sajeev and L. M. Nair, "LASER: A novel hybrid peer to peer network traffic classification technique," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2016.

[111] "jNetPcap," [Online]. Available: https://sourceforge.net/projects/jnetpcap/. [Accessed 2019].

[112] "Weka," [Online]. Available: https://www.cs.waikato.ac.nz/ml/weka. [Accessed 2019].

[113] P. Velan, M. Cermak, P. Celeda and M. Drasar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355--374, 2015.

[114] C.-N. Lu, C.-Y. Huang, Y.-D. Lin and Y.-C. Lai, "Session level flow classification by packet size distribution and session grouping," *Computer Networks*, vol. 56, no. 1, pp. 260--272, 2012.

[115] H. M. Sani, C. Lei and D. Neagu, "Computational complexity analysis of decision tree algorithms," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 2018.

[116] M. Dusi, F. Gringoli and L. Salgarelli, "Quantifying the accuracy of the ground truth associated with Internet traffic traces," *Computer Networks*, vol. 55, no. 5, pp. 1158--1167, 2011.

[117] "Global Internet Phenomena," Sandvine, 2019.

[118] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide and F. Jahanian, "Internet inter-domain traffic," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 75--86, 2010.

[119] J. Seibert, R. Torres, M. Mellia, M. M. Munafo, C. Nita-Rotaru and S. Rao, "The internet-wide impact of p2p traffic localization on isp profitability," *IEEE/ACM Transactions on Networking*, vol. 20, no. 6, pp. 1910--1923, 2012.

[120] S. Park, H. Chung, C. Lee, S. Lee and K. Lee, "Methodology and implementation for tracking the file sharers using BitTorrent," *Multimedia Tools and Applications*, vol. 74, no. 1, pp. 271--286, 2015.

[121] "List of TCP and UDP port numbers," [Online]. Available: https://www.wikiwand.com/en/List_of_TCP_and_UDP_port_numbers. [Accessed 2019].

[122] "Impact of COVID-19 on the Video Conferencing Market, 2020," [Online]. Available: https://www.businesswire.com/news/home/20200416005739/en/Impact-of-COVID-19-on-the-Video-Conferencing-Market-2020---ResearchAndMarkets.com. [Accessed May 2020].

[123] Q. Jiang, H. Hu and G. Hu, “Real-Time Identification of Users under the New Structure of Skype,” in *2016 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, 2016.

[124] Z. Yuan, C. Du, X. Chen, D. Wang and Y. Xue, “Skytracer: Towards fine-grained identification for skype traffic via sequence signatures,” in *International Conference on Computing, Networking and Communications (ICNC)*, 2014.

[125] S.-H. Lee, Y.-H. Goo, J.-T. Park, S.-H. Ji and M.-S. Kim, “Sky-Scope: Skype application traffic identification system,” in *2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2017.

[126] N. A. Saqib, Y. SHAKEEL, M. A. Khan, H. MEHMOOD and M. Zia, “An effective empirical approach to VoIP traffic classification,” *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 25, no. 2, pp. 888--900, 2017.

[127] S. Munir, N. Majeed, S. Babu, I. Bari, J. Harry and Z. A. Masood, “A joint port and statistical analysis based technique to detect encrypted VoIP traffic,” *International Journal of Computer Science and Information Security*, vol. 14, no. 2, p. 117, 2016.

[128] “Prepare your network for Meet video calls,” [Online]. Available: https://support.google.com/a/answer/1279090?hl=en. [Accessed September 2020].

[129] “Network Firewall Settings for Meeting Connector,” [Online]. Available: https://support.zoom.us/hc/en-us/articles/202342006-Network-Firewall-Settings-for-Meeting-Connector. [Accessed September 2020].

[130] “Ports need to be open to use Skype,” [Online]. Available: https://support.skype.com/en/faq/FA148/which-ports-need-to-be-open-to-use-skype-on-desktop. [Accessed September 2020].

[131] G. Sun, L. Liang, T. Chen, F. Xiao and F. Lang, “Network traffic classification based on transfer learning,” *Computers & electrical engineering*, vol. 69, pp. 920--927, 2018.

[132] F. Ertam and E. Avci, “A new approach for internet traffic classification: GA-WK-ELM,” *Measurement*, vol. 95, pp. 135--142, 2017.

LIST OF PUBLICATIONS

- **Max Bhatia** and Mritunjay Kumar Rai. "Identifying P2P traffic: A survey." Peer-to-Peer Networking and Applications, Springer (Published), 2017. (Scopus, SCIE 2.74 IF).

- **Max Bhatia**, Vikrant Sharma, Parminder Singh, and Mehedi Masud. "Multi-Level P2P Traffic Classification Using Heuristic and Statistical-Based Techniques: A Hybrid Approach." Symmetry (Published), 2020. (Scopus, SCIE 2.397 IF).

- **Max Bhatia** and Vikrant Sharma, "Classification of P2P File-sharing Traffic Using Heuristic Based and Statistical Based Technique." International Conference on Intelligent Circuits and Systems, CRC Press (Taylor & Francis Group), pp. 533-542, 2020. (Scopus).

- **Max Bhatia** and Vikrant Sharma, "Classification of P2P-VoIP (video) Traffic Using Heuristic-based and Statistical-based Technique" Conference on Global Emerging Innovation Summit, Bentham Science (Accepted), 2021. (Scopus).