uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Ahmad Hayajneh

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Electronic Business Technologies)

GRADE / DEGREE

School of Management

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Classification of Peer-to-Peer Traffic Using Data Mining Techniques and IP Layer Attributes

TITRE DE LA THÈSE / TITLE OF THESIS

Professor Bijan Raahemi

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

Professor M. Benyoucef

Professor D. Wright

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

uOttawa

L'Université canadienne
Canada's university

Classification of Peer-to-Peer Traffic Using Data Mining Techniques and IP Layer Attributes

By

Ahmad Hayajneh

Thesis Submitted to the
Faculty of Graduate and Postdoctoral Studies
In partial fulfillment of the requirements F
For the degree Master of Science in Electronic Business Technologies

Electronic Business Technologies
University of Ottawa

Thesis directed by:
Dr. Bijan Raahemi

Your file Votre référence
ISBN: 978-0-494-34074-5
Our file Notre référence
ISBN: 978-0-494-34074-5

Table of Contents

CHAPTER 1 - INTRODUCTION 1

CHAPTER 2 - P2P and ITS BUSINESS IMPACTS 6

2.1 P2P Evolution 6
2.2 Business Impacts on Telecom Network and Service Providers 10
2.3 Business Impacts on Production and Distribution (Copyright Issue) 20
2.4 Security Threats 22

CHAPTER 3 - RELATED PRIOR WORKS 24

CHAPTER 4 - RESEARCH METHODOLOGY 35

4.1 Research Design 35
4.2 Process and Methodology 37
4.3 Tools 39
4.4 Measurement Criteria 41

CHAPTER 5 - DATA PRE-PROCESSING 42

5.1 Data collection 42
5.2 Data Preparation 44

CHAPTER 6 - MODELING AND ANALYSIS 49

6.1 Modeling using Artificial Neural Networks 49
6.1.1 The Artificial Neural Network 49
6.1.2 Building the Models 52
6.1.3 Modeling analysis 58
6.2 Modeling using the Decision Tree 67
6.2.1 The Decision Tree 67
6.2.2 Building the Models 69
6.2.3 Decision Tree Modeling Analysis 73

CHAPTER 7 - CONCLUSIONS 79

7.1 The contribution of the research: 79
7.2 Future work 82
7.3 Publications resulted from this research 83

REFERENCES 84

List of Figures

Figure 1: Centralized Network Architecture. 7

Figure 2: Decentralized/ Distributed Network Architecture. 8

Figure 3: Hybrid Network Architecture. 9

Figure 4: Off-net Vs. On-net connection 11

Figure 5: Internet traffic analysis, June 2004. 12

Figure 6: Upload bandwidth by protocol. 14

Figure 7: Download bandwidth by protocol 15

Figure 8: P2P Traffic 16

Figure 9 : Global Mix of Peer-to-Peer Traffic. 18

Figure 10: Fixed offset detection 26

Figure 11: variable offset detection 27

Figure 12: {IP, port} pair heuristic 30

Figure 13: The elements of P2P pattern discovery 36

Figure 14: CRISP Data Mining Process Model 38

Figure 15. Data capturing setup 42

Figure 16: Sample prepared records 47

Figure 17: The data preparation process 48

Figure 18: The neuron functional model. 50

Figure 19: Backpropagation process 51

Figure 20 : Sensitivity vs. different number of records using four sets of attributes 57

Figure 21 : Specificity vs. different number of records using four sets of attributes 57

Figure 22 : Correctness vs. different number of records using four sets of attributes ... 58

Figure 23: Sensitivity vs. different P2P/Non-P2P records and various attributes..... 62

Figure 24: Specificity vs. different P2P/Non-P2P records and various attributes 63

Figure 25: Correctness vs. different P2P/Non-P2P records and various attributes..... 63

Figure 26: ROC curves for different Mix files..... 66

Figure 27: Binary decision tree 67

Figure 28: Ternary decision tree 68

Figure 29: Sensitivity vs. different number of records using two sets of attributes..... 71

Figure 30: Specificity vs. different number of records using two sets of attributes..... 72

Figure 31: Correctness vs. different number of records using two sets of attributes..... 72

Figure 32: Sensitivity vs. nine different P2P/Non-P2P ratios and Set #1 attributes' set 75

Figure 33: Specificity vs. nine different P2P/Non-P2P ratios and Set #1 attributes' set 75

Figure 34: Correctness vs. nine different P2P/Non-P2P ratios and Set #1 attributes' set

..... 76

Figure 35 : Time consumed to build the Neural Network (NN) and Decision Tree (DT)

models using 32011 training records..... 78

Figure 36: Neural network classifier..... 80

Figure 37: Decision tree classifier..... 81

List of Tables

Table 1: P2P applications and the embedded adwares. Source [23] 23

Table 2: P2P applications and their default port numbers 24

Table 3: Non-P2P internet applications using TCP and UDP concurrently 29

Table 4: The selected attributes for modeling phase 45

Table 5: The six extracted files with different number of records 47

Table 6: The four attribute sets considered in building the neural network models .. 52

Table 7: Confusion matrix 54

Table 8: The simulation results of 24 neural network classifiers with different number of records and four attributes' sets. 56

Table 9: Training sets with different mix of P2P and Non-P2P traffic. 60

Table 10: The 36 modeling results using neural network classifier, different number P2P/Non-P2P ratios of 3200 record file and four attribute sets. 61

Table 11: Attribute sets for modeling using J48 decision tree 69

Table 12: The 12 output classifiers using the J48 decision tree, different number of records and two attributes' sets. 70

Table 13: The 18 output classifiers using the J48 decision tree, nine different number P2P/Non-P2P ratios of 3200 record file and Set #1 attributes' set. 74

Acknowledgment

This research was supported by the Ontario Research Networks for Electronic Commerce (ORNEC) in collaboration with Research and Innovation Center of Alcatel-Lucent, Canada. I would also like to thank Peter Hickey of Computing and Communications Services (CCS), University of Ottawa, for his technical assistance in capturing traffic traces, and for several helpful discussions. Finally, I would like to thank the members of my thesis committee, Professor Morad Benyoucef and Professor David Wright, for accepting to review and comment on the thesis.

ABSTRACT

Peer-to-Peer (P2P) is an internet application that allows a group of internet users to share their files and computing resources. P2P traffic was tremendously increased to an estimated value of 70% of broadband traffic with a special nature that directly impacts the Telecom industry. Accordingly, the Telecom business has become very interested in finding solutions to identify and control P2P traffic.

This research focuses on developing a practical P2P traffic classification using data mining techniques and the information available in the TCP/IP header. We captured internet traffic, pre-processed and labeled them, and built several models using a combination of different attributes for various sizes of record files. We built the models based on neural network and decision tree techniques. Successful models were then subjected to a more stressful test using different ratios of P2P/Non-P2P in the training data set. We observed that the accuracy of the classification increases significantly when we take into account the source and destination IP addresses.

We concluded that source and destination IP addresses depict information about the "community of peers". Based on this observation, we recommended that the classifier needs to be implemented within the administrative domain of the individual service provider's network, and continuously updated to ensure that new communities of peers are detected, while old communities of peers are not penalized after they stop using P2P applications.

The proposed classification is based only on information in the IP layer, eliminating the privacy issues associated with deep packet inspection.

CHAPTER 1 - INTRODUCTION

Peer-to-Peer (P2P) is an Internet application that allows a group of internet users to communicate with each other, and directly access and download files (text, image, audio, and video) from the peers' machines. Also, it enables users to share their computers' resources, such as the processor and storage media, to build a distributed computing environment [1].

In 1997, ICQ offered a way to share files through its ICQ messenger using a file sharing service. In 1999, Napster introduced the first P2P file sharing tool that allowed peers around the world to download music files; within a short time, 60 million people around the world were using this application. Because of copyright legal issue related to the huge number of downloaded music files, Napster was sued and judicially closed, and the application is no longer available [2]. The development of P2P applications was continued and became easier to use. As a result, many P2P applications emerged including KaZaA, BitTorrent, EDonkey, WinMX, DirectConnect, LimeWire (Gnutella), eMule and more.

Problem Definition:

P2P traffic is generated when files and data are transferred between peers. Over the past few years, the P2P traffic has grown tremendously to an estimated value of 70% of the internet traffic [3] and [4]. P2P traffic and its characteristics have changed the original assumptions under which the data networks were designed. P2P traffic is more

symmetric (contrary to the assumption on which Asymmetric Digital Subscriber Line (ADSL) was designed); P2P traffic is less "bursty" which makes it difficult to take advantage of statistical multiplexing (under which the original data networks were designed). Also, P2P traffic lasts longer than typical web or email traffic, and packet lengths are mostly large, which keeps the queues in intermediate switches and routers more utilized, and consume more bandwidth and processing resources in the network devices. Finally, P2P traffic is less local and more spread among different autonomous systems spanning the globe. P2P applications utilize significant bandwidth and network resources, resulting in network congestion, affecting the availability, reliability and quality of services, and potentially reducing customer satisfaction. While allocating equipment for such significant network usage, telecom carriers and service providers do not gain proportional profits from the services they offer through their infrastructure.

P2P traffic impacts businesses on different sides. Even though carriers provide their networks for extraordinary usage by P2P traffic, they do not gain proportional profits out of this service. Practically, P2P traffic consumes the bandwidth and network resources which could result in network congestion that affects the availability, reliability and quality of services. This potentially impacts customer satisfaction and loyalty. Accordingly, ISPs are forced to upgrade their networks, and continuously spend on expanding the network resources. Also, P2P traffic has serious security issues related to the credibility of the contents of the shared files, and their integrities. Furthermore, there is a big debate regarding the rightfulness of the file sharing. As such, telecommunication equipment vendors and Internet Service Providers are interested in efficient solutions to classify and filter P2P traffic for further control and regulation.

Inline with the carrier's business goals, the control policy could be blocking the P2P traffic or billing the customer accordingly. Another policy could be to let the traffic flow in the network as long as it doesn't congest the network, and when resources (bandwidth, processing power, etc.) are limited, then block or limit the P2P traffic generated by the end users.

Early versions of P2P application were developed to use fixed TCP port numbers, which made it easy to identify P2P traffic. Later, when this kind of traffic considerably grew, ISPs were able to control it using the TCP port number. P2P software developers then produced new P2P applications that dynamically use different TCP port numbers to hide and mislead the ISPs traffic analyzers. In response, different approaches were proposed to detect P2P traffic. Unfortunately, most of these approaches encountered technical, business or legal issues. Very few others are promising but need more investigation and research development.

Research Objectives:

1. The main objective of this research is to develop an approach, using the data mining techniques, capable of classifying P2P traffic efficiently. Relying on the IP layers information, this approach will recognize the common patterns (network level characteristics) of the traffic generated by different P2P applications and find how P2P traffic is different from other internet traffic. The solution will determine the attributes that effectively describe different types of internet traffic, and how they are correlated to each others.

Also, the approach needs to be accurate and scalable to handle the vast amount of transactions occurring at high speed connections.

2. Furthermore, this research will investigate the business impacts of P2P networks on network and service providers, highlighting the interest of telecom industry in solutions to identify P2P traffic

Motivation:

This research is motivated by the following incentives:

1. Most of the new P2P identification approaches suffer from privacy issues or implementation barriers. Accordingly, through this research, we are looking for a new approach that addresses these problems.
2. Various telecom industries, such as carriers, equipment providers, content providers and third party solution providers, are interested in - and are already investing in - solutions to identify P2P traffic. Equipment providers will combine these solutions into their products and offer them to ISPs. ISPs will be able to regulate P2P flows traveling over their networks according to defined policies. They can restrict, throttle or allow P2P traffic as long as it does not congest their network, which enables them to guarantee Quality of Services (QoS) and comply with their Service Level Agreements (SLA). They will be able to redesign, upgrade and manage their networks to be scalable and reliable. Content providers will protect their business from unlawful distribution which advantages the society and economy.
3. A successful approach in detecting P2P traffic can be considered the base for classifying internet traffic for other purposes, such as intrusion detection, pattern

recognition, and trend analysis. Hence, new business opportunities for software developing firms in network monitoring, planning, and management could emerge.

Organization of the Thesis

The rest of this thesis is organized as follows: Chapter 2 explores the business impacts of P2P traffic on network and service providers. Chapter 3 discusses the approaches, features and limitations of the related P2P classification solutions already published in the literature. Chapter 4 introduces the research methodology including hypothesis, procedures and measurement criteria. Chapter 5 presents the research preparation phase where the data collection, processing and preparation are performed, and the modeling environment is set up. In Chapter 6, the modeling procedures are described, and the outputs results are analyzed. In Chapter 7, we conclude the research work and present our recommendations; we also discuss the potential future directions for this study.

CHAPTER 2 - P2P and ITS BUSINESS IMPACTS

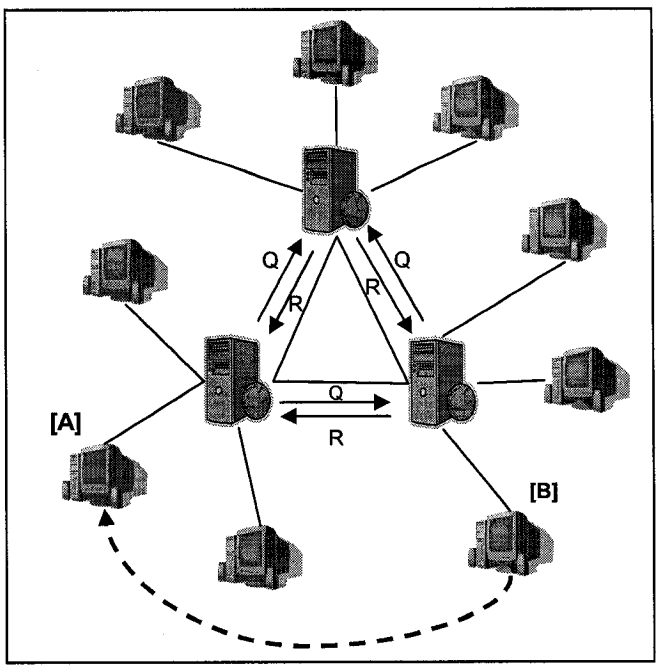
The fast proliferation and worldwide penetration of P2P software created serious difficulties for society and businesses, especially for the ISP and media production and distribution industries. In order to recognize how P2P traffic impacts these businesses, it is required to understand the evolution of P2P networks and their operational mechanism.

2.1 P2P Evolution

Since the first editions of P2P applications were introduced in 1999, their functionality has been extremely developed. Within the last 6 years, four generations of P2P networks were evolved as follows [5] and [6]:

1- First generation (Centralized Network)

The first generation architecture made use of a number of centralized index servers; each server maintains a database index of all directly connected clients and their shared files at any one time they are logged into the network. The database is updated whenever a client logs on/off the network. Figure 1 illustrates the operation of the centralized network.

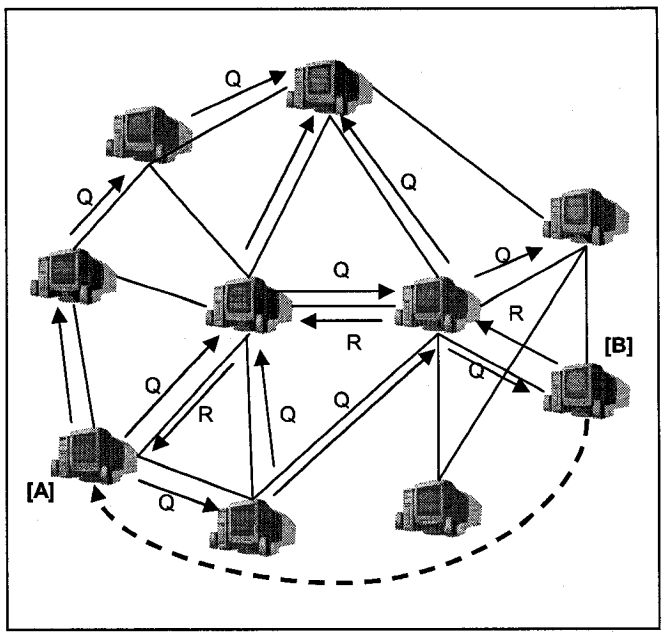


1- Node [A] sends a query to its local index server [1] searching a file.
2- Index server [1] checks if the requested file is available at any of the nodes directly connected to it, and disseminates the query to the other index servers on the network.
3- The query response from node [B] is returned to index server [1] which in turn forwards it to node [A].
4- Node [A] downloads the file directly from node [B].

Figure 1: Centralized Network Architecture. [Src: CasheLogic®]

2- Second Generation (Decentralized/Distributed Network)

P2P second generation networks have entirely adopted the decentralized/distributed architecture. Instead of central servers deployed on the network, any peer's PC acts as an integral part of the network performing the tasks of both the index server, for searching locally held resources and relaying queries between peers, and as a peer who shares resources. Figure 2 illustrates the operation of the decentralized network



1- Node [A] sends a query to the nodes it is directly connected to, searching a file.

2- Theses nodes check if the requested file is available locally and disseminates the query to the other nodes on the network.

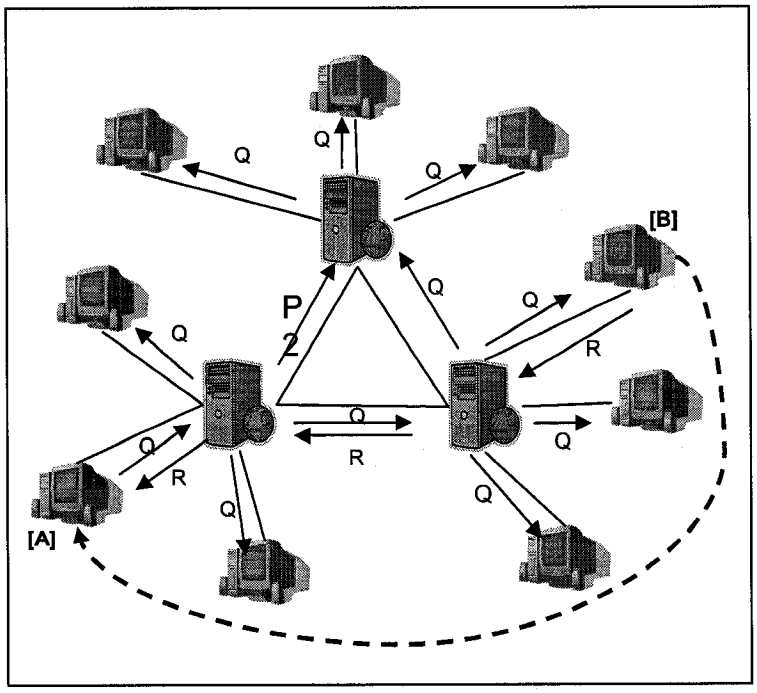
3- Node [B] locates the requested file on its drive and returns a response back across the network to node [A].

4- Node [A] downloads the file directly

Figure 2: Decentralized/ Distributed Network Architecture. [Src: CasheLogic®]

3- Third Generation (Hybrid Networks)

The third generation is hybrid architecture that is made up of the first and second generations. It combines the efficiency and resilience of the centralized network with the stealth characteristics of the decentralized network. This architecture deploys a hierarchical structure by establishing a backbone network of super nodes (or Ultra Peers) that also take on the duty of a central index server. When clients log on to the network, they makes a direct connection to a single super node which gathers and stores information about peer and content available for sharing, Figure 3.



1- Node [A] sends a query to the local super node [1] connected to searching a file.
2- Super node [1] checks if the requested file is available locally or at any of the nodes directly connected to it, and disseminates the query to the other super nodes on the network.
3- Node [B] responds to the query supernode [1] which in turn forwards it to node [A]
4- Node [A] downloads the file directly from node [B]

Figure 3: Hybrid Network Architecture. [Src: CasheLogic®]

4- Fourth Generation (Current Developments)

Recently, the new versions of P2P protocols are utilizing dynamic service port numbers in a range between 1024- 49151. But, in this range, there are so many port numbers that are customarily utilized by known non-P2P internet applications such as streaming (554, 7070, 1755, 6970, 5000, and 5001) or gaming (6112, 6868, and 6899). Accordingly, it is becoming increasingly difficult to know the true volume and nature of P2P traffic across service provider networks.

Further developments by the like of eDonkey and BitTorrent have seen the introduction of bi-directional streaming of traffic. Traditional architectures require a file to be fully downloaded before uploading can begin. The new P2P applications enable multiple and simultaneous downloads and uploads of a single file and/or multiple and simultaneous

downloads of an element of a single file. More clearly, they enable a single file to be downloaded by multiple subscribers and/or multiple subscribers to download the same file in fragments from multiple clients. This facilitates the rapid dissemination of content, as the more popular the item is, the more upload/download sources will be available from the moment the object begins to be downloaded. Eventually, a considerable increase in P2P traffic was noticed.

In summary, the evolution of P2P networks over these four generations made the file sharing faster to find, faster to download, globally adopted and extremely disguised. The Telecom industry was unprepared to handle this type of P2P. As a result, ISPs, and producers and distributors were quite impacted in different aspects.

2.2 Business Impacts on Telecom Network and Service Providers

P2P traffics impact ISPs in four key areas:

1- Increasing the bandwidth cost. Internet service providers consider various costs that can be allocated to their subscribers. One of the most significant costs is the charge of the service provider's internet transit connection (Off-net connection), the full connection from an end user at one ISP to another end user at another ISP through an international Exchange Carrier (ICX) such as MCI, AT&T or Sprint. Service providers pay the IXC for their internet transit bandwidth based on the total bandwidth they used. The more the subscribers use the service, the more it costs the service provider. This implies that the cost associated with serving each packet and connection depends on the location of an ISP subscriber's peer. Accordingly, crafting peering agreements with

other network providers reduces the amount of traffic and the cost of expensive internet transit connections. Local traffic (On-net), traffic that does not leave the service provider's backbone network, costs much less than traffic leaves the provider's domain (off-net), Figure 4.

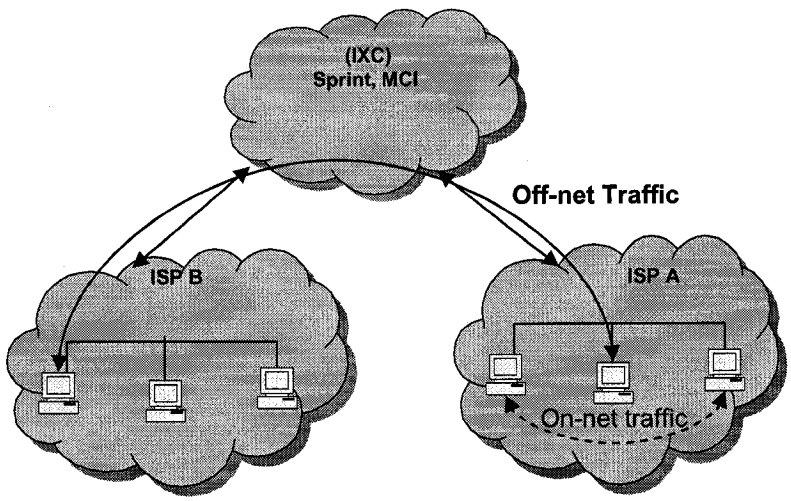


Figure 4: Off-net Vs. On-net connection.

Traditional uses of the internet were mainly On-net (email, Network News Transport Protocol [NNTP] or Web proxies), or destined for a small number of external content providers and data sites. With the global deployment of P2P applications, the On-net/Off-net usage was reversed to 80% / 20% Off-net/On-net traffic [7]. Also, prior to P2P, direct connections between home users were rare; P2P traffic significantly increases the amount of traffic between home users, especially with the increasing number of high speed subscribers. P2P file exchange increases the direct connections, which can be on-net and off-net, and as a result increases the bandwidth costs that cannot be easily passed onto consumers.

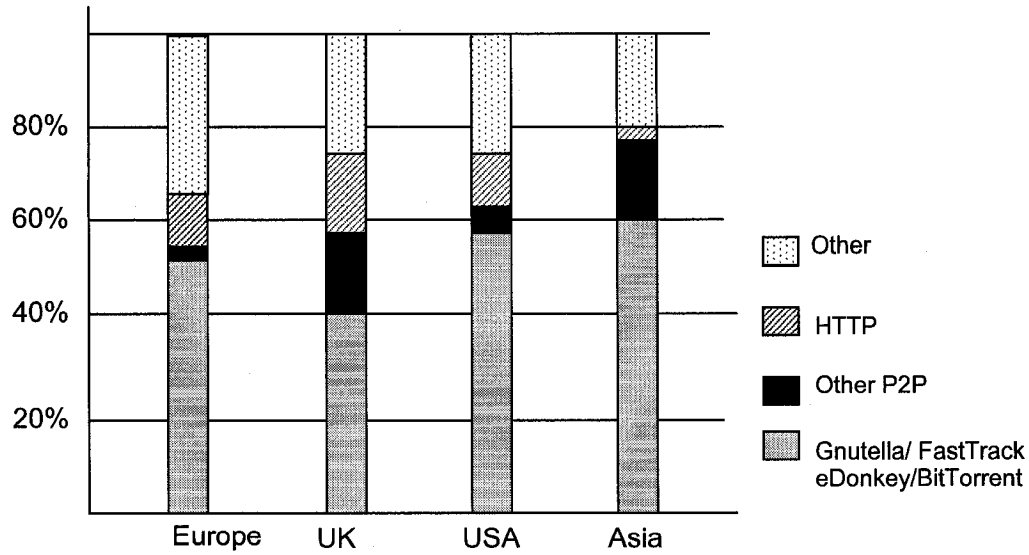


Figure 5: Internet traffic analysis, June 2004. Source [8]

Several market research surveys assure that P2P traffic contributes 70% of internet traffic [3], [4], [8], Figure 5. This vast percentage is interpreted as a large amount of dollars. Using simple statistics and calculations we can estimate how much P2P costs ISPs as follows:

Statistics:

Computer Industry Almanac Inc. estimated that:

World broadband internet users = 217,200,000 [9]

And,

the percentage of inter exchange P2P traffic = 70%of the total internet traffic

P.Cube ® estimated that Off-net traffic = 80% of the total Off-net traffic [7]

Practical Assumptions:

The average broadband package is 512Kbps/connection with 50:1 User/connection

ISP transit cost = 100 $US per Mb/s per month [10], [11]

Calculations:

Average transit per User = 512/50 = 10Kbps

Total transit = 217.2M X 10Kbps = 2,170.2 GB

Transit used of P2P = (70%) x (2,170 GB) = 1,520.4 GB

Off-net P2P traffic = (80%) x 1,519 GB= 1,216.32 GB

Cost of P2P transit = 1,216,320 X 100$

= 121.632 M$ per month

= 1.450 B$ annual by the end of 2005

As a result, ISPs pay IXCs 1.45B$ for their customers' P2P traffic. If we know that 10% of the ISPs' clients generate 90% of P2P traffic, we can understand why P2P costs ISPs a lot without even covering a small portion of their subscription cost. At the same time, ISPs can not penalize the other 90% of clients by forcing them to pay part of the P2P cost while they are not generating it.

2- Increasing infrastructure costs. Typical residential internet applications such as e-mail, Web browsing etc, generate a larger amount of downstream traffic for a single upstream request. Accordingly, service provider networks have been installed on a 10:1 downstream/upstream assumption ratio (asymmetrical) and optimized for the flow of data to the end subscriber. This ratio is correlated to the application requirements, network equipments with the attached interfaces, connections, processing and memories, and cabling infrastructure. If this assumption ratio is violated, then congestion, latency, freezing and unused capacity will result. However, P2P traffic is symmetrical (equal upstream/downstream ratio) in nature, with close to a 1:1 downstream/ upstream ratio - for every download there is a corresponding upload, Figures 6 and 7.

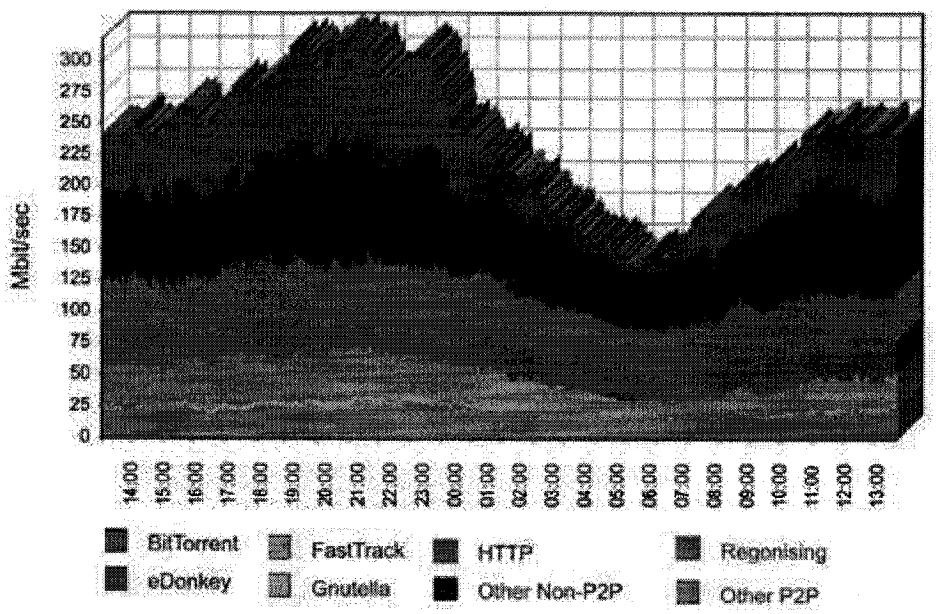


Figure 6: Upload bandwidth by protocol. Source [12]

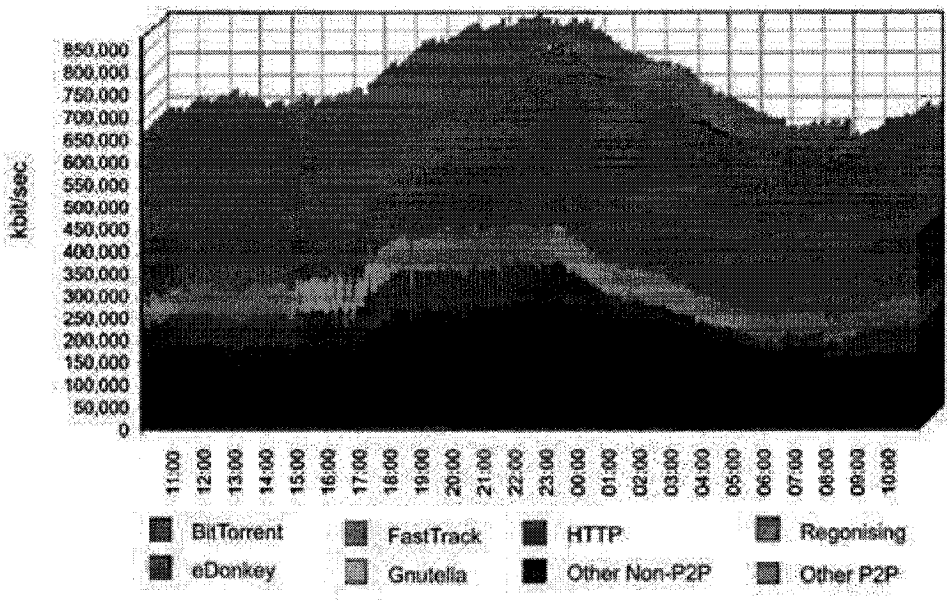


Figure 7: Download bandwidth by protocol. Source [12]

moreover, CashLogic®, a technology development company specialized in traffic management and network intelligence solutions, has shown that as much as 90% of the upload traffic over the last mile is P2P [8]. As a result, congestion occurs on the upstream link because of the larger number of subscribers using it. Physical attributes of the shared cable infrastructure Cable High-speed Data (HSD) providers are particularly limited in the amount of upstream network resources they have; they need to go through a costly configuration process (fiber node splits) to expand the capacity. Because P2P applications cause a dramatic increase in upstream data, they pose both cost and maintenance challenges for cable HSD providers. ISPs need to reconsider their network infrastructure design in order to cope with the exponential increase of P2P.

3- Network Congestion and Capacity Consumption: Service Providers experience increased congestion in their networks caused by the increasing volume of P2P traffic. Practically, P2P transaction is comprised of two main traffic types, [13]:

A. The connection management, which includes search queries/replies, signaling and keep-alive packets

B. The downloading traffic, which is the transfer of files between peers

Figure 8 illustrates the two main component of P2P network traffic

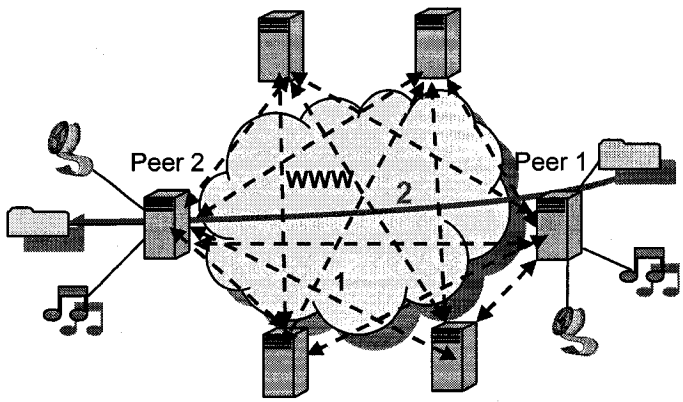


Figure 8: P2P Traffic

1: Connection management - - - - - -

2: Downloading file ——————

The connection management component of P2P traffic ensues from communication between different P2P hosts on the Internet. At the beginning, each peer uses a small packet (query) to search for the requested file, which results in many replies from peers who have that file. When two peers start to establish a connection, they exchange the

signaling packets using the TCP (SYN-SYN/ACK-ACK) handshaking. Also, at idle status, a number of small messages are communicated between all logged on peers, in order to keep the connection alive over a period of time, and ensure that file searches are quickly resolved.

Although the file downloads account for the significant percentage of the bandwidth, the connection management components also requires a notable share because of the millions of online peers searching, signaling or idling at a time. As many hosts leave their P2P connections running almost constantly, CashLogic estimates that there are always, at least, 10 million people logged on to a P2P network at any time [8]. The aggregate size of this component requires a substantial amount of the bandwidth.

A network bandwidth is limited for all its users, and certain oversubscription assumptions are considered when planning the capacity of the network. Traditional applications have a large "time-to-consume" factor, for example, an average webpage takes a few minutes to read, while an e-mail message might take a few seconds to process. The typical size of a music file is around 3-5MB, whereas movie files are at GB size. The average download speed of a normal high speed subscriber is around 150 Kbps, so the typical music file requires 20-35 second to download, while a movie of 1GB takes 1:50 hours. Figure 9 classifies P2P traffic into three categories, Video (61.44%), Audio (11.34%) and other (27.22%). This brings out the core problem of the P2P traffic nature; they are 73% of Video/Audio files. CashLogic reported that the vast majority of P2P traffic came from files in excess of 100MB and 30% of them were of 600MB, many of these were likely to be copies of films.

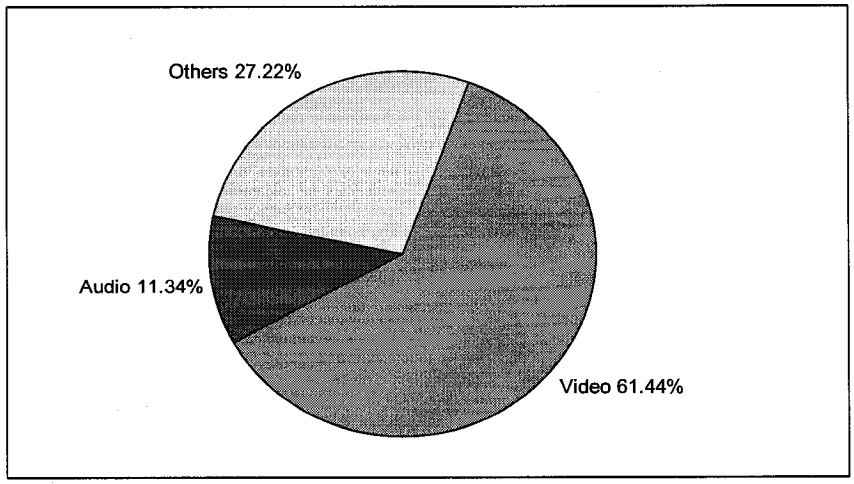


Figure 9 : Global Mix of Peer-to-Peer Traffic. Source [14]

The large size of Video/Audio traffic significantly consumes the bandwidth and network resources, affecting Quality of Services (QoS) parameters (throughput, packet loss, Jitter and delay), and making it difficult to honor the Service Level Agreement (SLA) between service providers and their customers, resulting in increased subscriber churn. This forces ISPs to continuously spend extra money to provide more network resources which is a costly and unpractical solution.

4- Network management problems: Based on subscriber profiling, service providers assume an average duration of network use/subscriber/day, as well as peak-use periods. Typically, service providers can predict and manage network "rush hours", as well as less-congested periods of network use. One important assumption is that residential home subscribers use the network primarily during weekends and at night, whereas telecommuters and small office or home office (SOHO) users are active during the business hours. Sporadic changes in these patterns may cause unplanned congestion.

Network monitoring detects that P2P applications are running 24 hours in the background, constantly downloading content, and they are left idle for many days. Providers have not factored these applications into time-of-day policies.

ISPs still lack the effective solution to manage and regulate P2P traffic; this is because they are unable to accurately identify P2P traffic in their networks. In fact, the fast and successful development in P2P applications within the last six years did not give the ISPs and there equipment providers enough time to develop P2P-controllers. Port number analysis, the most used method to control P2P traffic, becomes of no use with the introduction of the fourth generation of P2P applications. New approaches like Application Signature and Transport Layer Identification, as will be discussed in the next chapter, are still at the theoretical stage and suffer from serious limitations. The problem is becoming more aggravated; ISPs are susceptible by distribution firms and their associations to be sued for allowing copyright-infringing files to transfer over their networks, (RIAA vs. Verizon [15], RIAA vs. Charter Communications [16], and Bunt v. Tilley [17]).

As a result, because ISPs lack the solution to accurately identify P2P traffic, P2P users use ISPs' networks freely and for free. In addition, ISPs' network planning becomes in the windward. Furthermore, ISPs will no longer be able to manage and conform to their service agreements with their customers. Moreover, they may be troubled by other impacted businesses.

2.3 Business Impacts on Production and Distribution (Copyright Issue)

P2P file sharing is in the heart of the copyright and copy-fight battle; both parties defend their conception and justify their objective by the benefits of the society. This research does not discuss this argument nor advocates either party, but it presents both conceptions equally.

Copyright supporters are mainly the producers, distributors and their unions. They argue that P2P traffic is harming the software producers, distributors and customers because of the illegal file sharing. Authors have the moral and economic rights to benefit from their creations; distributors are investing a lot of money to publish a book, marketing and selling a film, song or software; theaters are paying thousands of dollars for a movie. Although there is a consensus on the damage, there are different opinions on the estimated value of this damage. First, recording companies and associations such as the Recording Industry Association of America (RIAA), the Canadian Recording Industry Association (CRIA), software developing firms and the Software & Information Industry Association (SIIA) allege that P2P file sharing caused billions in lost sales. CRIA general counsel, Richard Pfohl told a university audience that the Figure was around C$450M per year since 1999, totaling roughly C$2B over the past five years [18]. Graham Henderson, the president of (CRIA), argued that music downloading has devastated the industry [19]. McAfee reported that 36% or more than one in every three applications used in business is illegally used and P2P is mostly used to get this illegal software.

Some other researches concluded that even though the actual financial impact of music downloading has long been difficult to ascertain, the music industry was insignificantly

affected. Michael Geist [19] infers that the loss in Canadian artist sales that could be due to music downloading would stand at C$5.5 million per year. He concluded that reasons other than P2P are behind the declining sales on Canadian recording artists. At the same time, the International Federation of the Phonographic Industry (IFPI) reported that the surge in the global use of broadband is benefiting the legal music business, while illegal file-sharing remains virtually flat. Infringing music files available on file-sharing networks and websites raised only (3%), from 870M in January to 900M, while installed broadband links grew four times faster (13%) [20]. In fact, government legislations against copyright infringement were the main reason for the retreat of illegal file sharing. A research from IFPI suggests that there has been a clear shift of consumer attitudes in response to the well-publicized legal actions against file-sharers in 11 countries. More than 1 in 3 file-sharers surveyed in the US and the UK cites "fear of legal action" as the main reason for stopping illegal file-sharing [21].

On the other hand, freedom supporters (copy-fighters) argue that file sharing benefits the economy by creating new businesses opportunities such as anti-P2P software developers and research that employ many jobless people. Hence, if some businesses may loss some money because of file sharing, other industries emerge and revive, so the total economy and society will benefit. Furthermore, they claim that the market has to be open for all people without any restrictions, because this deregulation will fire the competition, which certainly incites humans to think and create.

2.4 Security Threats

P2P file-sharing allows a user to obtain and share any type of digital products. However, these applications are often abused by users with malicious intent to spread viruses, worms, spyware and other types of malware. "A security company TruSecure, through its division ICSA Labs, warned that there has been a significant surge in malicious codes posted on P2P networks. ICSA Labs officials claim that 45 percent of the thousands of free files they collected via KaZaA, the most popular P2P client, contained viruses, Trojan horses, and back doors" [22]

Some of the threats of the P2P may face a user while engaged in file-sharing include:

1 – Viruses, worms, Trojans etc. Certain viruses, worms, and spyware actually spread through P2P file-sharing networks. They usually masquerade as popular or benign files that entice a user to download and open them.

2 - Bogus files. Mostly these files originate from companies who release official products in order to fight piracy, or from some people who want to mock others or just to waste their time.

3 - Spyware/Adware. P2P programs are one of the most common gates through which an adware can infiltrate into users' computers and damage their computing experience. The adware can flood the PC with advertisements and pop-ups, hijack users' web browser, and slow their computers. Users have to be very careful of many P2P programs that claim to be free of charge. Table 1 lists some of P2P applications and the embedded adwares.

Table 1: P2P applications and the embedded adwares. Source [23]

P2P Program	Market Share	Adware/Spyware Installed
KaZaA	10.48%	Brilliant Digital, Gator, Joltid, TopSearch
Ares	2.73%	NavExcel Toolbar
Bearshare	2.58%	WhenU SaveNow, WhenU Weather
Morpheus	1.11%	PIB Toolbar, Huntbar Toolbar, NEO Toolbar
iMesh	1.01%	Ezula, Gator

4 - Missconfigured software. The internet is full of digital products that are malfunctioning. P2P users may download some applications from other peers that are missconfigured. When installed, the user will be lucky if these malicious codes cause improper functioning of their machine and he can uninstall them. Unfortunately, in most cases, these missconfigured applications would not be removed before damaging the users' PC configurations, and in many situations users are forced to format their PC drives to reconfigure their machine.

In brief, since there is no practical central authority that can verify the safety and integrity of the files shared, anyone can share anything, including files that the user did not wish to share. It is the responsibility of user to know how to use P2P application safely.

CHAPTER 3 - RELATED PRIOR WORKS

P2P traffic classification is part of the internet traffic classification conducted by internet business providers. ISPs rely on their network equipment providers' technologies to categorize the internet traffic type, including the P2P traffic. The TCP/UDP service port number was, and is still the most widely used method to identify the type of internet transactions. A tool, such as cflowd, Netflow, Netlog or others, reads the service port number in the TCP/UDP packet header, and if the port number matches with a predefined internet application port number, then the traffic is classified as that type of application. P2P developers have also assigned default port numbers to their applications. Table 2 lists the all P2P applications and their corresponding default port numbers we was able to find from different sources including the internet, technical reports and whitepapers.

Table 2: P2P applications and their default port numbers

Application	Service Port number
KaZaA	1214
BitTorrent	6881, 6889
Napster	6699, 6700, 6701
EDonkey 2000	4661, 4665
WinMX TCP/UDP	6257/6699
DirectConnect	411, 412
LimeWire (Gnutella Protocol)	6346, 6347
eMule TCP/UDP	4662/4672
Direct File Express	1044, 1045
WASTE	1337
CuteMX	2340
ShareDirect	2705
Abacast	4000-4100, 4500, 9000-9100

iMesh	4329
SongSpy	5190
Hotline Connect	5500-5503
Yoink	6666, 6667
Aimster/Madster	7668
BuddyShare	7788
Grouper	8038
hotComm	8080, 28864, 28864
Scour	8311
AudioGnome, OpenNap, Swaptor	8888, 8889
Blubster	41170

Even though these are the default port numbers, in reality, the new versions of P2P applications (the fourth generation) allow their users to dynamically use different numbers. Therefore, network analyzers become misled and unable to classify the P2P traffic, because either the analyzer does not know the utilized port before, or the port was already assigned for another type of internet application/traffic.

As a result, new methods were proposed to solve this problem. The proposed solutions addressed the P2P classification problem from different sides and layers; the application layer, the transport layer, statistical analysis of the network and transport information layer, and the communication behavior of peers.

The application signature approach was suggested by S. Sen, O. Spatscheck, and D. Wang in "Accurate, Scalable In-Network Identification of P2P Traffic using Application signature" [24]. The researchers relied on the fact that each internet application has a unique code (signature) that is attached to the data portion in the packet (payload). So, their objective was to develop a model for identifying the P2P traffic through finding and matching its application level signature. They examined the available documentations, which are scarce as P2P developers do not reveal such information, and packet analysis models to identify the application signatures, and then

they utilized these signatures to develop a scalable filter that can track P2P traffic on high speed internet connections for future detection.

For their study, they captured two packet traces from different network perspective points. The collected data was a valid set because it was representing the five well known P2P applications (KaZaA, Gnutella, eDonkey, DirectConnect and BitTorrent) which produce more than 95% of the P2P traffic. They decomposed P2P signatures into fixed pattern matches at fixed offsets within a TCP payload, or variable pattern matches with variable offsets within a TCP payload. For the fixed offset match they implanted a library function (byte match offset, word match offset, string match offset) that has a set of P2P signatures drawn from P2P documentation, Figure 10 illustrates the fixed offset match process.

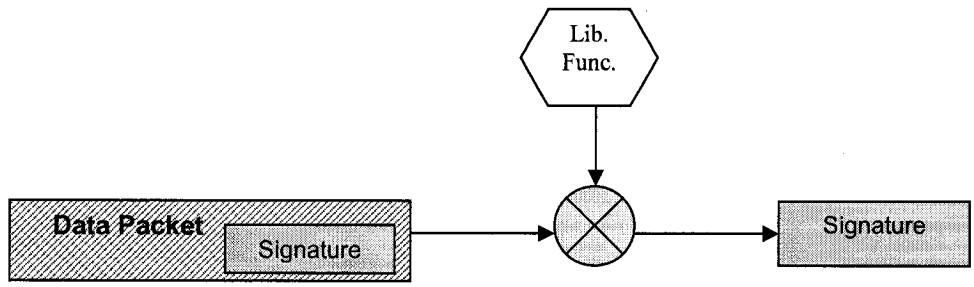


Figure 10: Fixed offset detection

The variable offset match was performed using some algorithms such as Standard Regex (SR), AST Regex (AR) or Karp-Rabin (KR) to search and find the variable string inside the payload, then they applied the output into the signature library for matching, Figure xxx illustrates the variable offset match process. Practically, this

process is expensive as it requires more processing time and resources. Figure 11 illustrates the variable offset match process

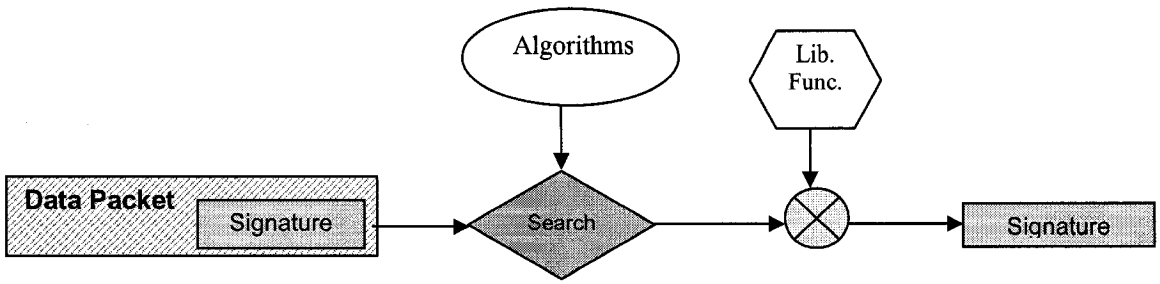


Figure 11: variable offset detection

The researchers evaluated the performance of the technique using accuracy, robustness and scalability measures. The results were promising. In terms of accuracy, this technique was able to achieve less than 5% of false negatives (misclassified P2P packets) for four P2P applications, and around 10% for BitTorrent. Also, it was capable of matching the signature by examining the first 6-8 packets of P2P traffic that promote this technique to handle high speed internet connections. On the other hand, such solution must consider the legal issues of its operational processes, and the usability obstacles at the business application level. This technique suffers from the two issues. First, from a legal point view, it is not allowed for an ISP or carrier to inspect the data portion of a packet because it includes secured, private and personal information of their customers which protected by law. Second, most of new P2P applications and the new versions of existing applications are using encryption algorithms to encrypt the data portion of the packet and consequently the signature, which makes it impossible to analyze the data and detect the signature.

Overall, the approach did not provide a workable solution, however it send a sign for other researchers to look for solutions that depend on information other than the data itself.

Another approach for P2P identification problem is named "Transport layer Identification of P2P traffic" suggested by T. Karagiannis, A. Broido, M. Faloutsos, and K. Klaffy [25]. Away from the signature analysis, this approach relies on the transport layer information and IP address vs. TCP port number relationship. The researchers analyzed data that was acquired from the Cooperative Association for Internet Data Analysis (CAIDA) and another data set that was captured from the University of Waikato. The data consisted of the first 44 bytes of each packet, which includes IP and TCP/UDP headers, and the first 4 bytes of the payload as they represents the P2P application unique string (signature). In their research, they monitored the nine most popular P2P protocols: eDonkey, Fasttrack, BitTorrent, OpenNap and WinMx, Gnutella, MP2P, Soulseek, Ares and Direct Connect. The solution was based on two heuristics; the first heuristic examined the source-destination IP pairs that use both TCP and UDP to transfer data (TCP/UDP heuristic). The second heuristic was based on how P2P peers connect to each other by exploring the connection characteristics of a {IP, port} pair.

Most of P2P applications use the UDP protocol for P2P session management, queries, query-replies and keep-alive packets, while the TCP is used for the actual data transfers. Six out of nine analyzed P2P protocols used both TCP and UDP in a single session. At the same time, this concurrent usage of both TCP and UDP is also used by other applications. The researchers examined all source-destination host pairs for which both

TCP and UDP flows exist. They found that besides P2P applications; only a few applications, listed in Table 3, use the same scenario also.

Table 3: Non-P2P internet applications using TCP and UDP concurrently

Application	Ports
NETBIOS	135,137,139,445
DNS	53
NTP	123
ISAKMP	500
Streaming	554,7070,1755,6970,5000,5001
IRC	7000, 7514, 6667
Gaming	6112, 6868, 6899
p2pnetworking.exe	3531

Accordingly, they stated that if a source-destination IP pair concurrently uses both TCP and UDP as transport protocols, then the traffic is considered P2P, so long as the source or destination port does not belong to one of the applications listed in Table 3. This bright observation is very helpful in building P2P classification solutions as it is common between most of P2P applications.

For the second heuristic, the P2P communication requires that when ever a user searches a file, he will advertise his {IP, Port} through the query packets in order for other peers to send him a response and upload the requested file. Accordingly, the researchers stated that for the advertised destination {IP, Port} of host A, if the number of distinct IPs connected to A is equal to the number of distinct ports that is correlated to it, then this traffic is considered to be P2P. They justified this by demonstrating that

when Peer A is downloading a file (or number of files) from different peers B, C, D, and E, then each peer will communicate with A using his own {IP, Port}. While if user A is browsing the internet and surfing many websites concurrently, then each web server will communicate with user A using the {IP $_{server}$, 80} pair. Figure 12 illustrates this justification. This observation is good and can help to solve part of the problem as it explored the idea of IP vs. Port relevance.

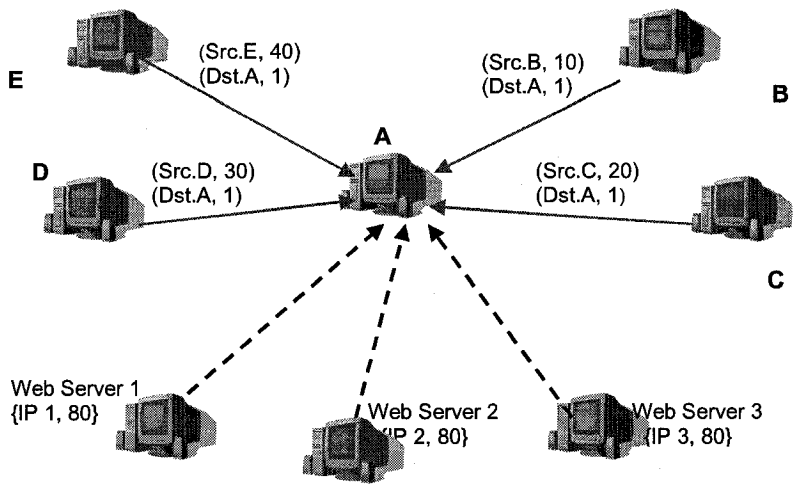


Figure 12: {IP, port} pair heuristic

The researchers claim that their approach was able to detect more than 95% of P2P traffic on an operational OC48 backbone link, while the false positive ranges approximately from 8% to 12% which is good results to achieve.

Altogether, this approach introduced new ideas to look at the P2P traffic, however, it experiences two main deficiencies related to the first and second heuristics. First, the

solution can not identify a P2P application that is using any port number listed in Table 3. Second, the second heuristic will be useless if a peer is downloading a file from another one peer or different files from different peers, each file from a single peer, while surfing the internet, sending emails to his friends and downloading a file from FTP server at the same time. As a result the approach is good from some special and limited usage.

A similar concept of the transport layer identification was adopted by T. Karagiannis, K. Papagiannaki and M. Faloutsos in "BLINC: Multilevel Traffic Classification in the Dark" [26] to classify general internet traffic. To make the solution more powerful, the researchers integrated the host behavior with the UDP/TCP utilization pattern. They argue that observing the activities of a host will provide more information and can reveal the nature of the users' applications which is very true and another good approach to consider but to some extend that will not violate the customers' privacy. The user behavior is captured at three different levels: 1) Social, where it examines the popularity of a host, and identifies communities of nodes which may correspond to clients with similar interests or members of a collaborative application. 2) Functional, where the behavior is captured in terms of its functional role in the network, namely whether it acts as a provider or consumer of a service, or both, in case of a collaborative application like P2P. 3) Application level, where the transport layer interactions between particular hosts on specific ports are captured with the intent to identify the application of origin. The classification was provided using only four attributes (source IP address, destination IP address, source port, and destination port). Then, it was refined further by exploiting other flow characteristics such as the UDP/TCP utilization

behavior and the average packet size, finally integrating the hosts' behavior using the sport number. For different data sets, the effectiveness of the solution was high (80%-90% of internet traffic types were classified with an accuracy around 90% - 95%). In fact this effectiveness is understandable as long as the approach relies on the port service number to form different levels of user behavior. As a result, it expected that this approach will work with most internet traffic types that have static port number. However, the P2P traffic that always changes its port number. Also, the authors consider their classification work as a statistical problem, and they built their solution on data sets of 455GB, 1223 GB and 1652GB, which are very huge file sizes that infer sufficient information. However, they did not did not mention the computational tools or the processing time it took to achieve these results, so there is a question about the usability of the approach.

Novel emerging researches discuss the behavior and characteristics of internet traffic using different statistical approaches. In "Self-learning IP Traffic Classification based on Statistical Flow Characteristics" [27], S. Zander, T. Nguyen, and G. Armitage proposed a framework for IP traffic classification based on statistical flow properties using an unsupervised machine learning (ML) technique. They utilized the Sequential Forward Selection (SFS) approach to identify the optimal set of flow attributes that minimize the processing cost while maximizing the classification accuracy. In this research, the authors first classified packets into flows according to IP (sour, des) addresses, port numbers and protocols using NetMate. Then they used the attributes: packet inter-arrival time, packet length, mean and variance, flow size, and session duration, without justifying why these attributes were selected, to build the ML

classifier model. They used the Autoclass for ML classification, an implementation of the Expectation Maximization (EM) algorithm. EM is an unsupervised Bayesian classifier that automatically learns the 'natural' classes (also called clustering) inherent in a training dataset with unclassified cases. While the authors planned to evaluate their approach using a larger number of flows and more applications, and experimented with more attributes, they confessed that the precision of the resulting classifier and the classification performance had not been evaluated. Even thought the researches did not reasoned why they used the Bayesian classifier not others, the ML utilization still a valid solution with more examination.

Equally, and without just looking fro the P2P traffic only, Denis Zuev and Andrew W. Moore in their paper "Traffic Classification using a Statistical Approach" [28] proposed a supervised machine learning approach to classify network traffic. They capitalized on data that had been previously hand-classified, and allocated traffic to one of ten predefined categories: Bulk, Database, Interactive, Mail, WWW, P2P, Service, Attack, Games and Multimedia. The authors utilized 248 per-flow discriminators (characteristics), such as flow duration, TCP Port, packet inter-arrival time (mean, variance, etc), payload size (mean, variance, etc), effective Bandwidth based upon entropy, and more. To build their model, they used the basic algorithm of Naive Bayes analysis using the Weka data mining toolkit. Then, they enhanced the algorithm using the kernel density estimation theory. Furthermore, they considered the Fast Correlation-Based Filter (FCBF) for best feature selection and redundancy reduction, in order to improve the classifier accuracy and reduce time processing. They evaluated the performance of the solution results in terms of accuracy (the raw count of flows that

were classified correctly divided by the total number of flows) and trust (the probability that a flow that has been classified into a class, is, in fact, from this class). They showed that in its most basic form, the Naive Bayes classifier was able to provide 66.71% accurate results, but it can achieve 93.5% accuracy if combined with the kernel density estimation theory, and up to 96.3% if FCBF was incorporated also. Although this approach was promising, there is a question about the performance, and accordingly the usability of the classifier. This is because too many discriminators were used. Therefore, the authors acknowledged that more research is needed to search for the best possible discriminators. Both last two approaches relay on Naive Bayes analysis which considers the value of the values of the utilized attributes alone without understanding the relationship at higher levels, which will lead to a failure in the solution functionality.

Eventually, the proposed solutions tried to build an accurate, workable and scalable classifier for P2P traffic. Although each approach addressed the problem from single or two points of view, these approaches provide new valuable perspectives of looking into the P2P problem classification. At the same time, their limitations must be a good experience for the coming researches in order not to be trapped.

CHAPTER 4 - RESEARCH METHODOLOGY

Internet traffic is composed of different protocol types such as Web surfing (HTTP), Email (SMTP, POP or IMAP), File download (FTP), file sharing (P2P), Voice over IP (VoIP), Streaming and many more. Each Internet traffic type has its own distinct pattern at the application or network level. These patterns can be described by different attributes such as service port number, packet length, Quality of Services (QoS) parameters (network parameters), application signature (application level parameter), or by communication behavior such as the use of UDP and TCP protocols concurrently in the communication session. This study capitalized on the network level information, the packet arrival time, packet ID, packet length, protocol type, source IP address and the destination IP address to build a classifier that determine the P2P traffic pattern.

The research capitalized on P2P and Not-P2P traces that were collected from the university gateway. The data was prepared and then subjected to different data mining techniques to discover the behavior of the P2P traffic. The discovered behavior was analyzed in terms of network layer attributes and their correlations. Different resulted classifiers were validated using accuracy measures.

4.1 Research Design

Throughout this research, we will find out what attributes contribute efficiently in describing the P2P behavior and how these attributes are related to each other in order to recognize the P2P patterns.

Focus of the research: this research focus on utilizing the data mining techniques to analyze real internet data and understand its correlations in order to infer valid P2P patterns. Data mining, also known as knowledge discovery of Databases, is the finding of meaningful hidden patterns from large data sets using automated statistical and correlation analysis techniques. Data mining was originated to work with a huge size of data sets, and built with sophisticated statistical calculations, logical comparison, neural simulation and correlation analysis. At the same time, data mining proved successful achievements in other different fields such as customer relationship management, marketing, banking, fraud detection, etc. As its main function, data mining was able to discover patterns hidden inside raw data. Figure 13 illustrates the elements of P2P pattern discovery.

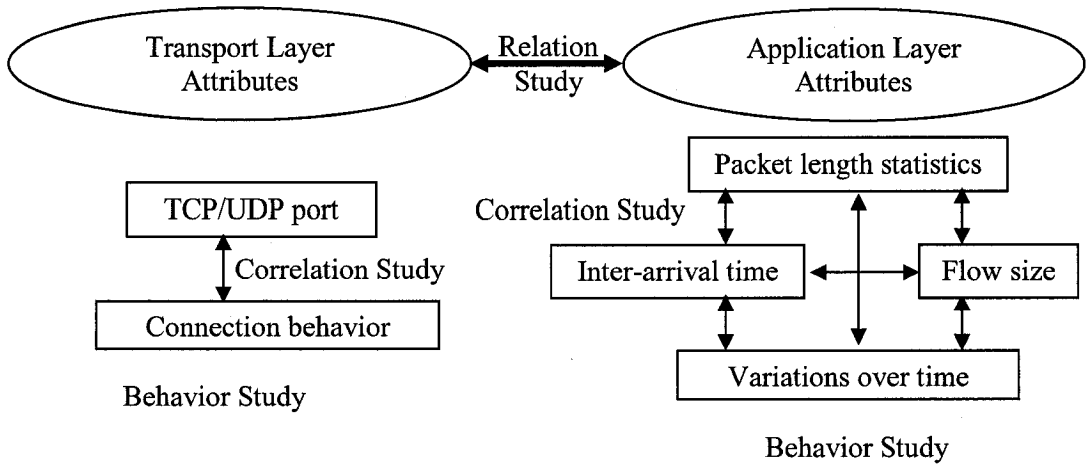


Figure 13: The elements of P2P pattern discovery.

Data size: The examined data files were in Mega Bytes, and the number of records varies from 2000 to 64000 instances. This number of records is necessary for different data mining techniques to learn P2P behavior and validation purposes.

Data type: The collected data includes the packets header of the network and transport layers only because this research intends to build a classifier that relies on data that can not be masked or violate personal privacy. The data was reduced to represent P2P and Non-P2P traffic only. This data was mixed of numerical and nominal types; at the same time, nominal data was converted to a numeric value to fit some technique requirements.

Data analysis and modeling: data was entirely analyzed by the Weka 3.4 data mining suit using the decision tree and the artificial neural networks classification approaches [29]. This research relay on the classification task to discover the P2P pattern and develop a classifier that is able to identify P2P traffic among other internet traffic. Classification is a supervised learning approach that learns the classes of data from a pre-classified data (training set) and maps future data into these classes. Also this study will examine the clustering technique for P2P pattern discovery and traffic identification. Clustering is the grouping of data based on similarities without a pre-training of the data groups (classes). However, judgments and conclusions were mixed of human justifications and modeling outputs.

4.2 Process and Methodology

The core process of this research adopted the CRoss Industry Standard Process for Data Mining (CRISP-DM) model [30]. (CRISP-DM) is the industry standard methodology

for data mining and predictive analysis. Figure 14 illustrates the main stages of CRISP-DM.

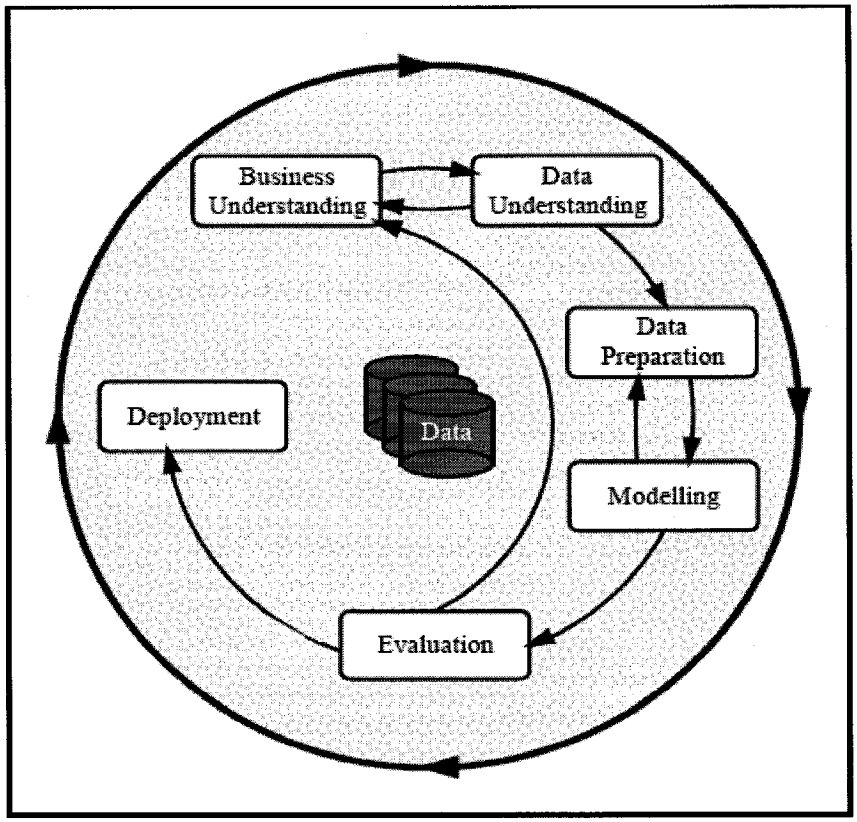


Figure 14: CRISP Data Mining Process Model

According to the CRISP process, we have divided our research work in to three main phases:

1 - The "Business Understanding" Phase, where project objectives and requirements are clearly specified in terms of the problem or research objective, and the goals are translated into the formulation of a data mining problem definition. At this stage, Data capturing, preparation, and analysis tools are installed, configured and practiced.

2- *Data Pre-Processing* which includes:

2.1- The "Data Understanding" Phase. In this phase raw data is collected, categorized and overviewed to gain initial insights into data; then the quality of this data is evaluated and, if required, interesting subsets that may contain useful information are selected.

2.2- Then the "Data Preparation" Phase is applied where the final data set is prepared from the initial raw data. The cases and variables are selected to analyze the data. If required, transformations are performed on certain variables, and the raw data is cleaned so that it is ready for the modeling phase.

3- *Building the model* which includes:

3.1- The "Modeling" Phase, where appropriate modeling techniques are applied, best attributes are selected and model settings are calibrated to optimize the results. If necessary, this phase loops back to bring the form of the data into line with the specific requirements of a particular data mining technique.

3.2- Then the "Evaluation" Phase is used to evaluate the resulted models for quality and effectiveness before being deployed in the field. During this phase, it is determined whether the model in fact achieves the objectives in the first phase. The final decision is made depending on the measurements of results.

The "Deployment Phase", where the developed models are deployed in the field for real utilization is out of the research's scope.

4.3 Tools

Throughout this research, we used the following tools for data capturing, date preparation and modeling processes:

1- *Tcpdump* [31], A Unix-based platform tool/service that was developed by Lawrence Berkeley Laboratory Network Research Group. It is an open source command-line tool for monitoring (sniffing) network traffic. Tcpdump is a powerful tool that can capture and display packet headers and match them against a set of criteria. It understands Boolean search operators and can use host names, IP addresses, network names, and protocols as arguments. Because it is free, powerful and easy to use, Tcpdump was used to capture real internet traffic from the university gateway router.

2- *Windump* [32] is the Windows® ported version of the UNIX Tcpdump. Windump is fully compatible with Tcpdump and can be used to watch and diagnose network traffic according to various complex rules. Because the data analysis of our research was undertaken on Windows based computers, Windump was utilized to process data files that were captured by Tcpdump.

3- *PHP (Hypertext Preprocessor) language* [33] is a reflective programming language originally designed for producing dynamic Web pages. PHP is used mainly in server-side application software, but it can be used from a command line interface or in standalone graphical applications. PHP was used to develop a code for records labeling.

4- *MS Excel* is a spreadsheet program written and distributed by Microsoft for Windows operating system computers. MS Excel was used to organize data in tables to facilitate data manipulation, cleaning and subseting.

5- *Weka Data mining toolkit* [34]. It is a collection of machine learning algorithms for data mining tasks; it was developed at the University of Waikato in New Zealand. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is an open-source software issued under the GNU General

Public License. Weka was used in this research because it is a non-commercial tool that is originated for research objectives, and it includes a large number of data mining techniques under different algorithms, in addition to a big set of data filters and attribute- selection features.

4.4 Measurement Criteria

The following measurement criteria were used to evaluate different models:

1- *Accuracy*; It is the degree of trueness of the developed classifier in classifying input records. It is evaluated using the *Correctness, Sensitivity* and *Specificity* measures.

1.1- *Correctness* is the coincident of the total classified instances to the total actual instances, and it is measured by the percentage of the correctly classified number of instances to the total number of instances.

1.2- *Sensitivity* is the coincident of the classified P2P instances to the total actual P2P instances, and it is measured by the percentage of the correctly P2P classified number of instances to the total number of P2P instances.

1.3- *Specificity* is the coincident of the classified Non-P2P instances to the total actual Non-P2P instances, and it is measured by the percentage of the correctly Non-P2P classified number of instances to the total number of Non-P2P instances.

2- *Performance*: In this research, the performance was measured by the time consumed by different data mining techniques to learn the P2P pattern (build the model).

CHAPTER 5 - DATA PRE-PROCESSING

Based on the CRISP data mining process model (see section 4.3), the data pre-processing consists of Data Collection and Data Preparation phases.

5.1 Data collection

Data was collected using the Tcpdump service in the university computing and communications services (CCS) center. The data collection process was undertaken over five days at different time periods in April 2006. In total 37 files with different sizes (2-15) GB were generated, totaling 250GB of binary format data. The captured data was entirely composed of the TCP/IP packet headers of two-way internet traffic. The data was a good representation of the ISP internet traffic as it included traffics from student residents, academic and business (administrative) units. Figure 15 shows the general configuration of our setup to capture the traffic.

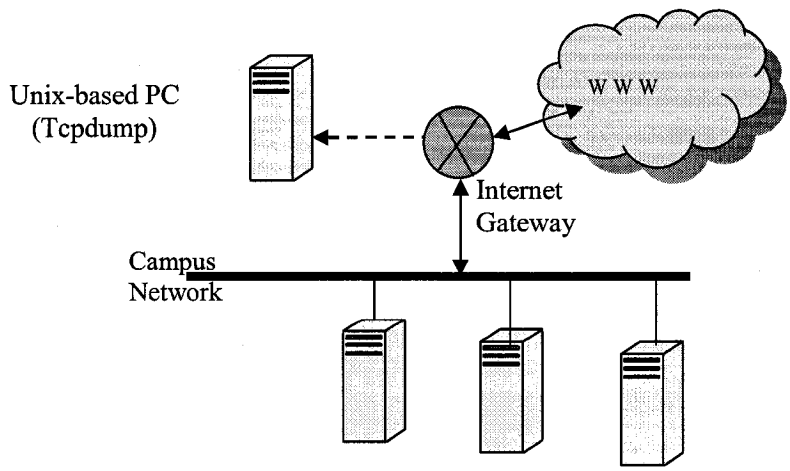


Figure 15. Data capturing setup

All files were labeled and stored on a safe storage media for further processing.

Using Windump, these file were filtered and transferred into readable text format as follows:

```
Read Tcpdump_file {port-number ls 1024 OR Eq {1214,
6881, 6889, 6699, 6700, 6701, 4661, 4665, 4672, 4662,
6346, 6347, 6348, 6349, 6257, 1044, 1045, 1337, 2340,
2705, 4500, 4329, 5190, 5500, 5501, 5502, 5503, 6666,
6667, 7668, 7788, 8038, 8080, 28864, 8311, 8888, 8889,
41170, 3074, 3531} write File_name.txt
```

The port numbers less than 1024 represent the non-P2P traffic, while all other port numbers represent all P2P applications we found throughout our research, Table 1.

As a result, two types of data were collected, pure Non-P2P and pure P2P entries. It is worth noting that several P2P applications were inspected to confirm that they enable users to randomly select the port number in the range of 1024 – 65535 only, and none of them allow users to use a port number less than 1024. The following entries represent a sample of the collected data,

```
04:38:15.297248 IP (tos 0x0, ttl 63, id 31129, offset 0, flags
[none], proto: UDP (17), length: 264) 137.122.70.23.4921>
81.66.28.51.6346: UDP, length 236

04:38:15.298019 IP (tos 0x0, ttl 105, id 48442, offset 0, flags [DF],
proto: TCP (6), length: 1500) 81.66.28.51.6346 > 137.122.70.23.4921: .
3732432837:3732434297(1460) ack 1274371307 win 65081
```

```
04:38:15.298116 IP (tos 0x0, ttl 127, id 34354, offset 0, flags [DF],
proto: TCP (6), length: 40) 137.122.73.47.2075 > 216.120.241.17.80: .,
cksum 0xc292 (correct), ack 2920 win 65535
```

5.2 Data Preparation

In this phase data files were prepared to an understandable and usable format; they were subjected to the following preparation steps:

1- The text formatted data was applied to a PHP code that separated each field in the packet with a comma separator.

2- *Transformation.* Data was converted to a spreadsheet format using MS Excel®. Initially each file has 66050 records (packet).

3- *Cleaning, Filtering, Attribute selection.* The initial inspection of the spreadsheet with a simple statistic analysis showed that:

- Very few records (519 = 0.7%) have missing or scrambled data, so these entries were discarded. (Cleaning)
- The "tos", "flag" and "offset" fields have almost one single value for all records (unary attributes), and "ttl", "win", "cksum" and "Sequence number" contain no information to differentiate records (non-informative attributes), so these entries have non-distinguishing information so they were all dropped off also. (Filtering)
- The "Arrival time", "Identification", "protocol", "Packet Length", "Source IP", "Destination IP", "Source Port" and "Destination Port" fields have different values that can be utilized to discriminate records (informative attributes). Accordingly, they were retained. Table 4 summaries the selected attributes and their description.

Table 4: The selected attributes for modeling phase

Attribute	Description	abbreviation
Arrival time	The time that the packet arrive at the gateway.	"Arr. Time"
Identification	Used to identify the fragments of one datagram from those of another. The originating protocol module of an internet datagram sets the identification field to a value that must be unique for that source-destination pair and protocol for the time the datagram will be active in the internet system.	"ID"
Protocol	This field specifies the encapsulated protocol.	"Protocol"
Packet Length	The length of the Packet (40 - 1500) Byte.	"Length"
Source IP	IP address of the sender.	"Src. IP"
Destination IP	IP address of the intended receiver	"Des. IP"
Source Port	A number that donates to IP network service that the client use to request from the other end, server or peer.	"Src. Port"
Destination Port	A number that donates to IP network service that the client use to deliver to the other end, server or peer.	"Des. Port"

The "Arrival time" solely does not provide any useful information, but it will be valuable when studying the time within each entire communication session. The "ID" value is set by the originating protocol module of the internet datagram and it must be unique for that {source-destination} pair and protocol for the time the datagram will be active on the internet. Again, if it is solely monitored it will not provide any useful information, but if it is studied within an entire session it will infer the sequence of the

packets for the session. The "protocol" was retained because this research will benefit from the first conclusion of [8] stated that most P2P applications used TCP and UDP as transport protocols concurrently. The "Packet Length" is important because it was noticed that most of P2P packet lengths, regardless of the application, were around 1500B. The "Source IP" and "Destination IP" will be correlated to other fields in order to form a complete useful behavior of the P2P traffic. Finally, the "Source Port" and "Destination Port" will be used for labeling the entries in the next step, as both ports were was utilized to filter them. (Attribute selection).

4- *Labeling*. All data records were labeled as being P2P or non-P2P traffic using the following pseudo code. The actual code was written in PHP language. In this step the port number was used for labeling purposes only in order to generate the required instances for training the classifier and validating it; as such, the port number would not be used for building the classifier itself.

```
If (Src. port number OR Des. port number) < 1024
   Then Type = "Non-P2P"
Else
   If (Src. port number OR Des. port number) > 1024
    Type = "P2P"
End
```

As a result several files of 65351 records each were resulted with (xls), (CSV) and (txt) format. Figure 16 shows a sample labeled records with various fields in a resulted file.

Time	ID	Protocol	Length	Src. IP	Des IP	Src Port	Des Port	Type
04:38:15.297248	31129	17	264	137.122.70.23	81.66.28.51	4921	6346	P2P
04:38:15.298019	48442	6	1500	81.66.28.51	137.122.70.23	6346	4921	P2P
04:38:15.298116	34354	6	40	137.122.73.47	216.120.241.17	2075	80	Not-P2P

Figure 16: Sample prepared records

5- *Filing*. For the modeling process (next chapter), six different files with different number of records were extracted from one arbitrary selected file as shown in Table 5. The records were randomly extracted. The purpose of these files was to know what is the best number of records that a classifier needs to learn the P2P pattern, and to accurately classify P2P traffic.

Table 5: The six extracted files with different number of records

File	Number of Records	P2P/Non-P2P records
F1	2007	461/1546
F2	4008	952/3056
F3	8001	1979/6022
F4	16025	4820/11205
F5	32011	10371/21640
F6	65531	20577/44954

Figure-17 illustrates the data preparation process explained in the preceding section.

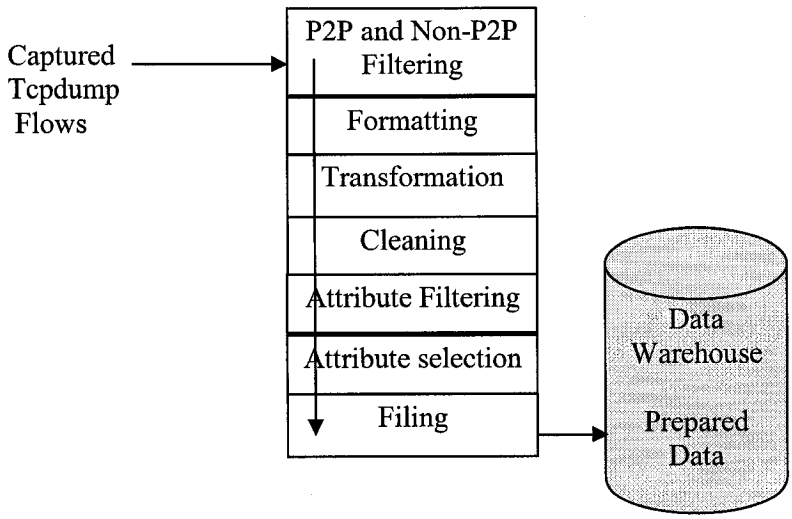


Figure 17: The data preparation process

CHAPTER 6 - MODELING AND ANALYSIS

As the core of this study, different models of P2P traffic classifiers were built using the decision tree and neural network algorithms. During the modeling process, it was important to build the classifiers with the optimal configurations that assure the best classification results. Modeling phase was carried out on a P4, 3.0 GHz CPU, 512 RAM and 100GB hard drive Windows® based machine located in the Knowledge Discovery and Data Mining Laboratory (KDD Lab) at the School of Management, University of Ottawa.

6.1 Modeling using Artificial Neural Networks

6.1.1 The Artificial Neural Network

The Artificial Neural Network (ANN) [35], or simply Neural Network (NN), is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. Neural networks, like people, learn by example. A neural network is configured for a specific application, such as pattern recognition or data classification, through a learning process. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons; this is true of the neural network as well. Neural networks can be used to extract patterns and detect trends that are too

complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. Figure 18 illustrates the neuron fundamental function.

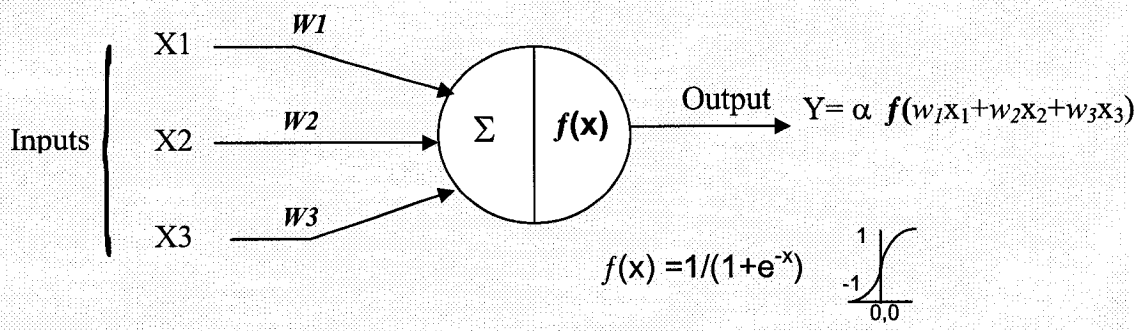


Figure 18: The neuron functional model.

In this study, we used the MultilayerPerceptron model [34] using the backpropagation algorithm, the neural network technique supplied by Weka. The backpropagation algorithm, an abbreviation of the backward propagation of errors, is a supervised learning technique that basically modifies the weights of the connections leading to the hidden layer nodes (neurons) based on the strength of each node's contribution to the final predication. It is the most useful approach for feed-forward networks (networks that have no feedback/loop back connections). The backpropagation requires that the transfer function used by the artificial nodes be differentiable. The process of the technique is summarized as follows:

1. Apply input (training sample) to the neural network (Xi).

2. Calculate the output (Y).

3. Compare the resulting output with the desired output (actual) for the given input. This is called the *error (E)*.

4. Modify the weights (w_i) for all neurons using the *error*.

5. Assign a threshold for each node function, giving greater responsibility to nodes connected by stronger weights.

6. Repeat the process until the error reaches an acceptable value (e.g. error < 1%), which means that the NN was trained successfully, or if we reach a maximum count of iterations, which means that the NN training was not successful.

As the algorithm's name implies, the errors propagate backwards from the output nodes to the inner nodes. So, backpropagation is used to calculate the gradient of the error of the network with respect to the network's adjustable weights. Figure 19 illustrates the backpropagation mechanism.

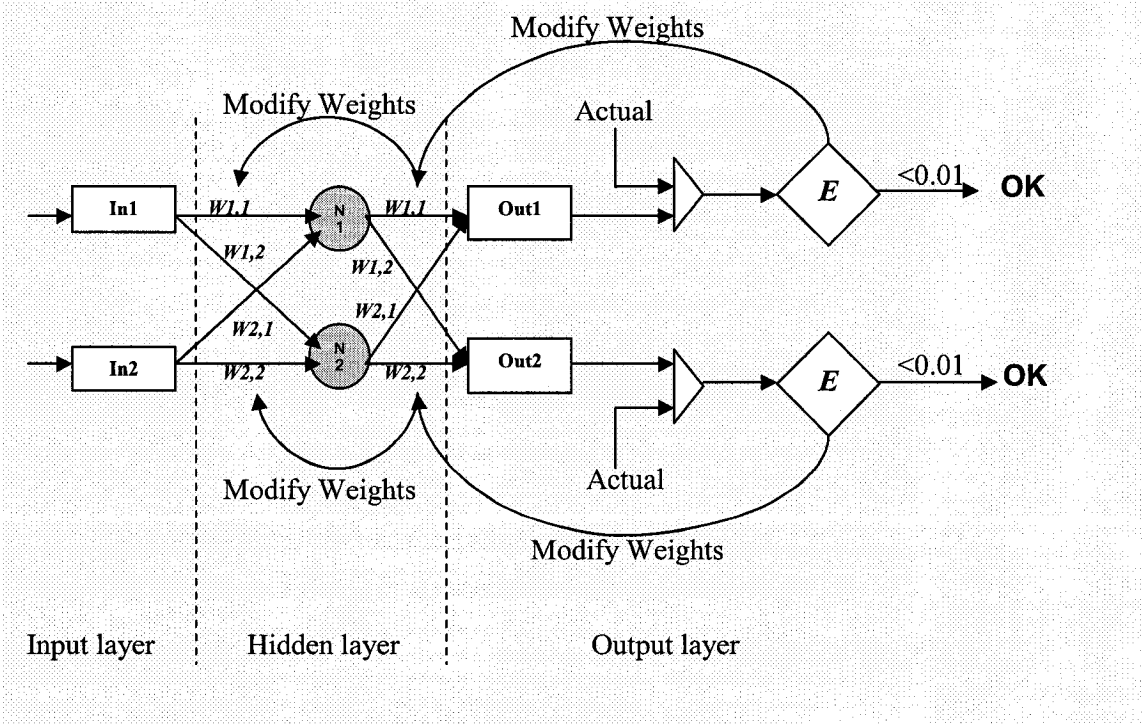


Figure 19: Backpropagation process

6.1.2 Building the Models

For building a classifier using the neural network, four sets of attributes were made from the selected attributes in Table 4 in order to practically select the best attributes that effectively describe the P2P pattern, Table 6 describe these sets.

Table 6: The four attribute sets considered in building the neural network models

Attribute Set	Attributes
Set #1	"Arr. Time", "ID", "Protocol", "Length", "Src. IP", "Des. IP"
Set #2	"Arr. Time", "Protocol", "Length", "Src. IP", "Des. IP"
Set #3	"Protocol", "Length", "Src. IP", "Des. IP"
Set #4	"Arr. Time", "Protocol", "Length"

In order to improve the neural network performance, the attributes "Src. IP" and "Des. IP" were converted into its equivalent numerical values. For the Arr. time", it was noticed that all records in each file have the same hour and minute time value, so hour and minute portions were discarded, and only the second and the micro-seconds time portions were considered. The attributes "Src IP" and "Des IP", were originally captured in the string decimal dotted format, they were converted into their equivalent numerical value using the following formula:

$$IPnum = A*(256^3) + B*(256^2) + C*(256) + D \qquad (1)$$

Where A.B.C.D is the dotted decimal representation of the IP address.

As all attributes become numeric, they were normalized into a value between [0 , 1] using the Normalize filter in Weka. This will significantly improve the neural network performance process.

Then the neural network was configured as follows:

1. *Autobuild*: True. This is to add and connect up hidden layers in the network if needed; the default is one hidden layer.
2. *Decay*: True. This will cause the learning rate to decrease. This will divide the starting learning rate by the epoch number (number of passes the network will take through the data) in order to determine what the current learning rate should be. This may help to stop the network from diverging from the target output, as well as improve the general performance of the technique.
3. *HiddenLayers*: i. One hidden layer with nodes equal to the number of input attributes.
4. *LearningRate*: 0.3. this is an initial value and it is updatable.
5. *Monmentum*: 0.2. The momentum applied to the weight during updating.
6. *RandomSeed*: 0. Seed used to initialize the random number generator. Random numbers are used for setting the initial weights of the connections between nodes, and also for shuffling the training data.
7. *Reset*: True. This will allow the network to reset with a lower learning rate. If the network diverges from the answer this will automatically reset the network with a lower learning rate and begin training again.
8. *TrainingTime*: 500. The number of epochs to train through.

9. *TestOption*: Cross-Validation Fold 5. The original sample is partitioned into 5 sub samples. A single sub sample is retained as the validation data for testing the model, and the remaining 4 sub samples are used as training data. The process is then repeated 5 times (the *folds*), with each of the 5 sub samples used exactly once as the validation data. The 5 results from the folds then are averaged to produce a single estimation

The neural network model we built in this study was to classify the input traffic into two groups of P2P and Non-P2P classes (i.e. binary classifier). Accordingly, a single instance has four different prediction outputs, two are correct which are: (a) True Positive (TP) where an instance is actually P2P and it is classified as P2P; and (b) True Negative (TN) where an instance is actually Non-P2P and it is classified as Non-P2P; and two false predictions: (c) False Positive (FP) where an instance is classified as P2P but actually it is Non-P2P; and (d) False Negative (FN) where an instance is classified Non-P2P but actually it is P2P. These are visually represented in the confusion matrix of Table 7.

Table 7: Confusion matrix

		Predicted class	
		P2P	Non-P2P
Actual class	P2P	(TP)	(FN)
	Non-P2P	(FP)	(TN)

Relying on the confusion matrix, the accuracy of the resulted classifier was evaluated using the *Sensitivity, Specificity* and *Correctness* measures.

- Sensitivity, also called the True Positive rate, is measured by the percentage of the correctly P2P classified number of instances to the total number of P2P instances.

Sensitivity = truly classified P2P records / all P2P records

$$Sensitivity = TP / (TP + FN) \qquad (2)$$

- Specificity, also called the True Negative rate, is measured by the percentage of the correctly Non-P2P classified number of instances to the total number of Non-P2P instances.

Specificity = truly classified Non-P2P records / all Non-P2P records

$$Specificity = TN / (FP + TN). \qquad (3)$$

- Correctness is measured by the percentage of the correctly classified number of instances to the total number of instances.

Correctness = truly classified (P2P and Non-P2P) records / all data records

$$Correctness = (TP+TN)/ (TP+TN+FP+FN) \qquad (4)$$

The higher the values of Sensitivity, Specificity and Correctness, the more accurate the classifier is. The study is looking for a solution model that has at least 90% of Correctness, Sensitivity and Specificity measures each.

The six input files with different number of records F1 - F6 (prepared in data pre-processing phase) were applied to the neural network using the four attribute sets in Table 6. As a result, 24 classifiers were built. Table 8 shows the resulted models.

Table 8: The simulation results of 24 neural network classifiers with different number of records and four attributes' sets.

File	Attributes' Sets	Time to Build (Sec)	Correctness (%)	TP rate (%) Sensitivity	TN rate (%) Specificity	FP rate (%)
F1 2007 record	Set#1	7	80.4	57.9	85.5	14.5
	Set#2	6	79.1	54	84	16
	Set#3	4	45.3	22.3	51.2	48.8
	Set#4	3	36.8	17	41	59
F2 4008 records	Set#1	14	85.5	60.2	90.2	9.8
	Set#2	11	83.8	52.3	90	10
	Set#3	9	40.3	25	45.2	54.8
	Set#4	6	41.1	21.3	50.4	49.6
F3 8001 records	Set#1	29	86	63.7	93.2	6.8
	Set#2	22	83.3	54.9	93	7
	Set#3	18	47.7	26.6	56.2	43.8
	Set#4	13	52.7	27.8	61	39
F4 16025 records	Set#1	59	87.1	70.9	95.3	4.7
	Set#2	45	83.6	57.9	96.5	3.5
	Set#3	37	56.1	35.2	66.3	33.7
	Set#4	27	53	28.4	65.3	34.7
F5 32011 records	Set#1	119	88.7	75.5	95.4	4.6
	Set#2	92	86.6	65.3	96.8	3.2
	Set#3	76	80.1	59.1	90.9	9.1
	Set#4	55	74.3	43.4	89.8	10.2
F6 65531 records	Set#1	240	94.5	87.2	98.2	1.8
	Set#2	185	94.1	86.3	98	2
	Set#3	154	92.6	82	97.5	2.5
	Set#4	111	88.6	73	96.5	3.5

Figures 20 to 22 illustrate the Correctness, Sensitivity, and Specificity measures of Table 8

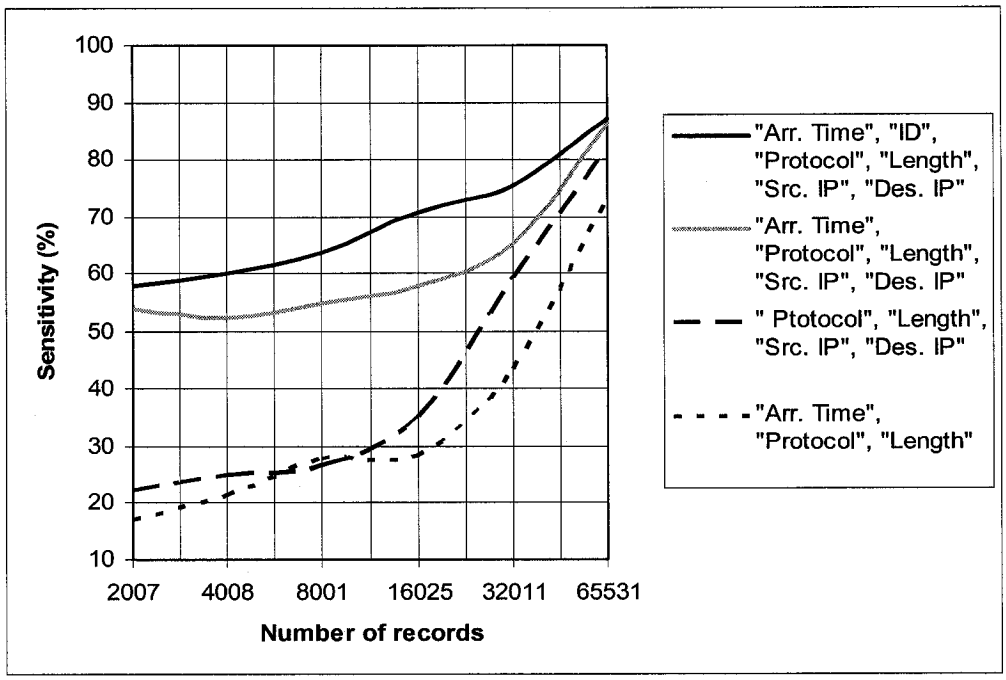


Figure 20 : Sensitivity vs. different number of records using four sets of attributes

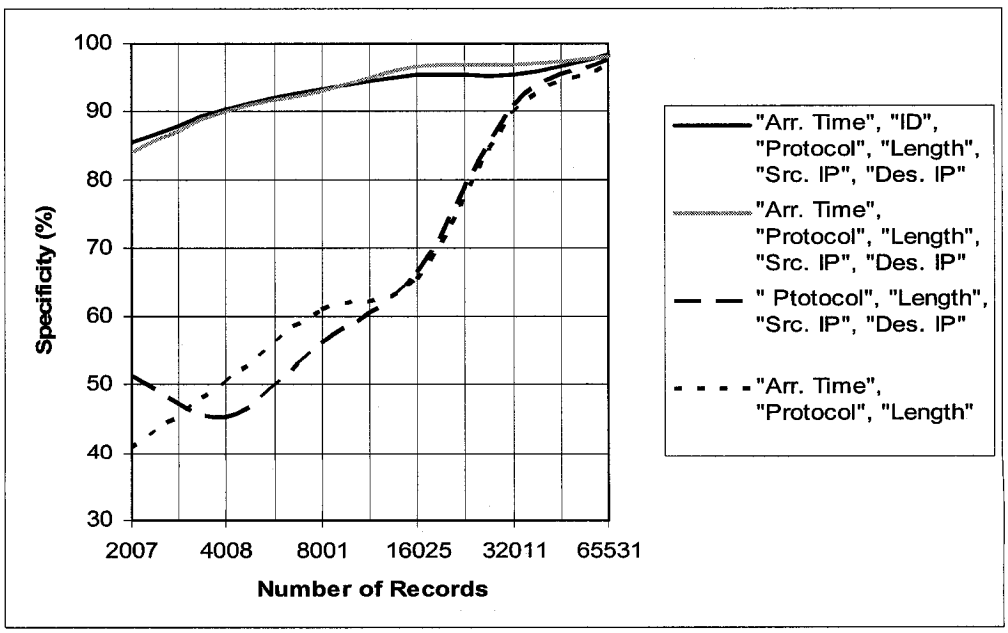


Figure 21 : Specificity vs. different number of records using four sets of attributes

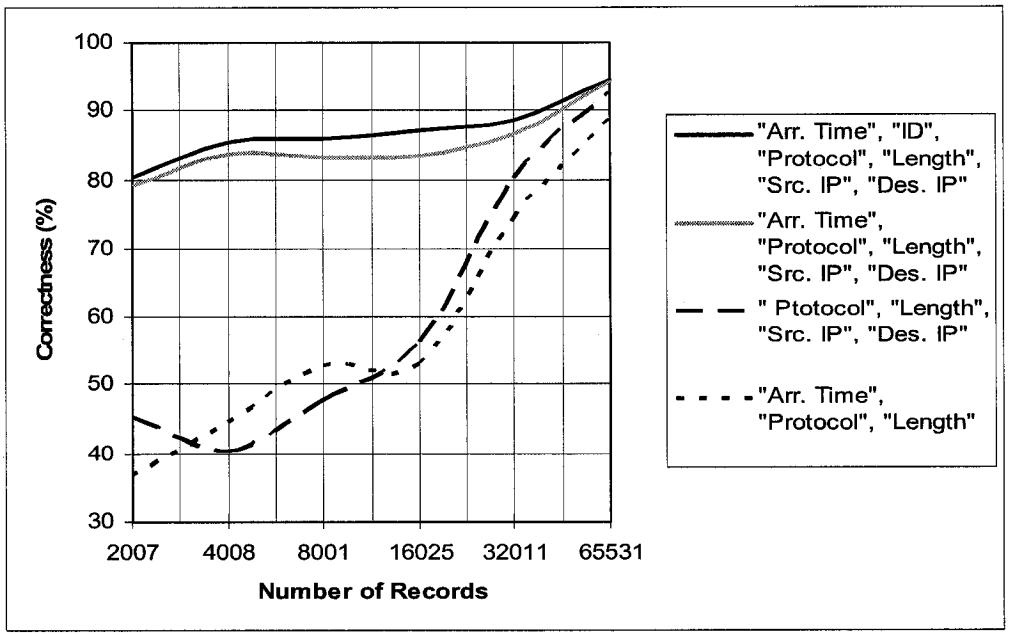


Figure 22 : Correctness vs. different number of records using four sets of attributes

6.1.3 Modeling analysis

The modeling results and the corresponding charts demonstrate that:

1- The neural network was able to identify the P2P traffic correctly (>90%) when the number of records was 65531 records, at which the number of P2P records was 20577 (Sensitivity chart).

2- The Neural network succeeded to learn the Non-P2P traffic and produce a high Specificity rate (>90%) even for a small number (4008) of Non-P2P records (Specificity chart).

3- Removing the attributes "Arr. Time" or "Src IP" and "Des. IP" degraded the accuracy of the classifier significantly.

We elaborate on these observations with more details. The Non-P2P traffic that we captured belonged to the internet applications with port numbers less than 1024, which are mainly www, email, network management (SNMP), FTP and few others. These applications establish short sessions with few numbers of packet exchanges. Consequently, in the captured traffic file with thousands of records, there is a large enough number of Non-P2P packets - representing complete sessions- with the required time sequence and successive ID values to train the neural network model. This is confirmed by the good Specificity rate (>85%) for 2007 and 4008 records, and almost sustained high values of the Specificity (>95%) for a large number of records (>8000) that include sufficient numbers of Non-P2P traffic to properly train the neural network model.

On the other hand, P2P traffic includes packets generated by the transmission of large files (image, audio or video files). As such, a large number of P2P records are required to keep track of successive IDs as well as the sequence of time within a session in order to properly train the neural networks. This was also confirmed by the low Sensitivity measures (<75%) for all files when a small percentage of P2P records was included in the training set. However, for a large enough number of P2P records (20577 in file # 6) the classifier was trained well, and the Sensitivity rose to 86% (when including "Arr. Time", "Src IP" and "Dst. IP").

The Correctness provides an overall trueness of the classifier condition that the classifier has the enough information about all classes to train. Because the Correctness is directly related to Sensitivity and Specificity; so, while the Sensitivity has low values with small enough number of P2P records, and the Specificity has high values and

represents the majority of the traffic, the reasonable values of the Correctness (>80%) did not indicate a reasonable functionality of the classifier.

In order to confirm our justification and find a threshold of P2P records for which the neural network model performs the best, we prepared nine sample files, Mix1 to Mix9. Each file includes 32000 records with different ratios of P2P/Not-P2P records; Table 6 shows these files and the corresponding number of records ratios.

Table 9: Training sets with different mix of P2P and Non-P2P traffic.

File	P2P/NotP2P records
Mix1	3200 / 28800 = (10/90)%
Mix2	6400 / 25600 = (20/80)%
Mix3	9600 / 22400 = (30/70)%
Mix4	12800 / 19200 = (40/60)%
Mix5	16000 / 16000 = (50/50)%
Mix6	19200 / 12800 = (60/40)%
Mix7	22400 / 9600 = (70/30)%
Mix8	25600 / 6400 = (80/20)%
Mix9	28800 / 3200 = (90/10)%

Using the same neural network configurations and the four attribute sets, the different mix files of P2P/Non-P2P traffic were used to build new models. As a result, 36 classifier models were built. Table 10 shows the resulted classification models.

Table 10: The 36 modeling results using neural network classifier, different number P2P/Non-P2P ratios of 3200 record file and four attribute sets.

File	Attributes	Time to Build	Correctness (%)	Sensitivity (%)	Specificity (%)
Mix1	Set #1	119	93.8	71.3	96.3
	Set #2	92	93.8	56.9	97.9
	Set #3	75	91.2	23.4	98.7
	Set #4	55	87.0	18.3	94.6
Mix2	Set #1	120	88.3	76.6	91.2
	Set #2	93	86.9	54.9	94.9
	Set #3	76	86.7	41.2	98.1
	Set #4	55	78.0	22.7	91.8
Mix3	Set #1	121	82	64.6	89.5
	Set #2	93	82	48.6	96.2
	Set #3	75	80.9	52.1	93.2
	Set #4	55	72.1	35.0	88.1
Mix4	Set #1	119	82.9	74.8	88.3
	Set #2	91	84.5	71.9	92.9
	Set #3	72	78.4	68.3	85
	Set #4	55	66.3	42.9	78.3
Mix5	Set #1	119	87.6	85.5	89.8
	Set #2	92	88.3	80.3	96.4
	Set #3	76	75.8	72.2	79.3
	Set #4	55	68.8	62.9	74.8
Mix6	Set #1	118	91.7	91.7	91.6
	Set #2	92	92.1	88.9	96.9
	Set #3	75	75.2	78.9	72.6
	Set #4	55	68.3	71.9	64.0
Mix7	Set #1	119	93.4	93.1	94
	Set #2	92	93.3	91.6	97.1
	Set #3	74	77	80.4	70.7
	Set #4	55	68.3	72.2	60.4
Mix8	Set #1	120	96.3	96.8	94.3
	Set #2	93	95.5	95.7	94.8
	Set #3	76	77.8	82.2	55.9
	Set #4	55	72.3	76.1	53.8
Mix9	Set #1	120	97.8	98.3	93.1

	Set #2	93	97.9	98.3	94.1
	Set #3	76	79.4	86.5	50.2
	Set #4	55	73.5	77.0	45

Figures 23-25 illustrate the Correctness, Sensitivity and Specificity measures of Table 10

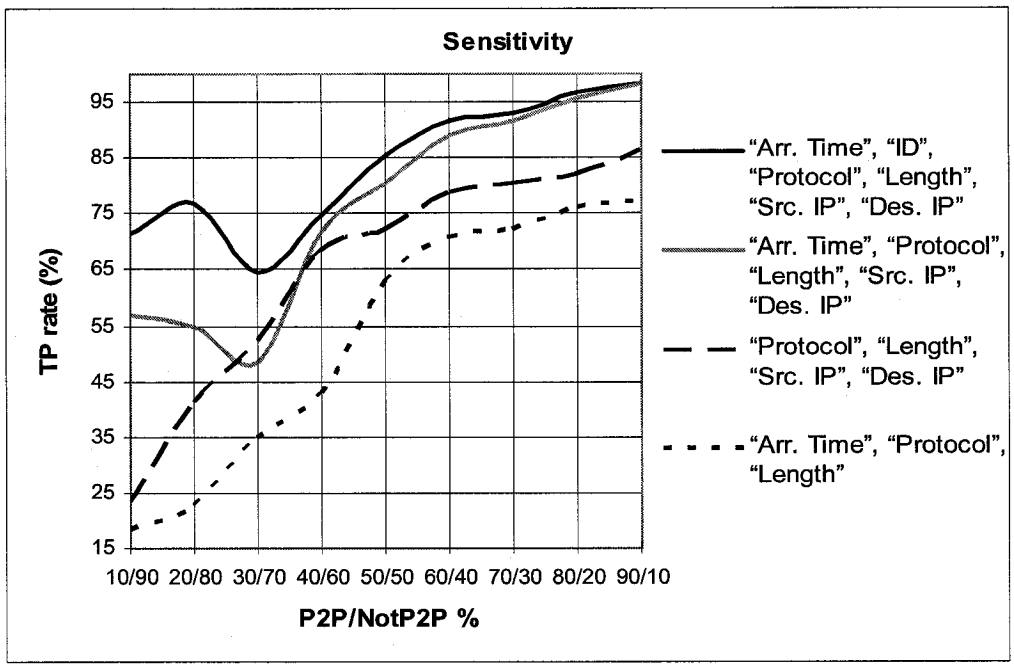


Figure 23: Sensitivity vs. different P2P/Non-P2P records and various attributes

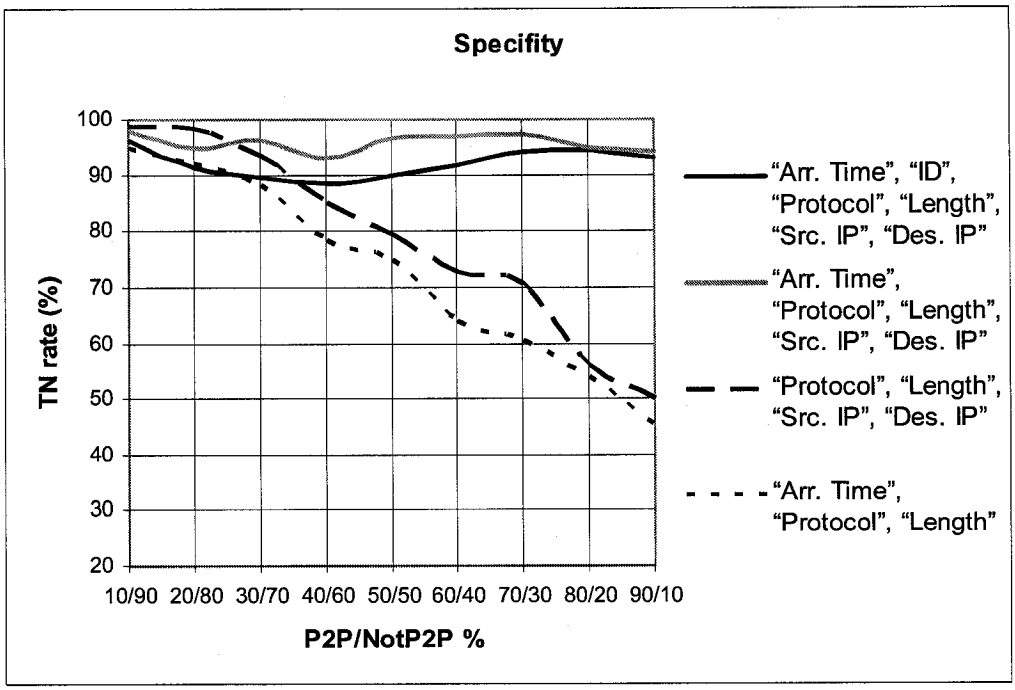


Figure 24: Specificity vs. different P2P/Non-P2P records and various attributes

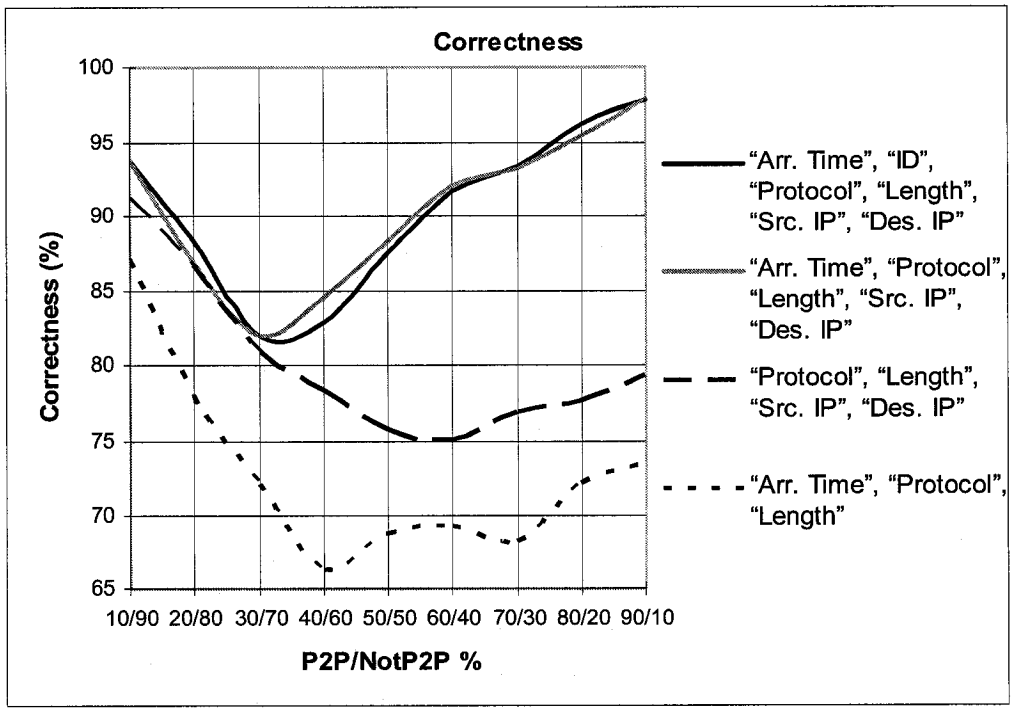


Figure 25: Correctness vs. different P2P/Non-P2P records and various attributes

The modeling results and the corresponding charts confirm that the accuracy depends on two factors: the selected attribute set, and the ratio of P2P/Non-P2P traffic. In particular:

1. Above a certain threshold of P2P records (>50/50 = 16000), the accuracy of the classifier increases proportionally with the increase of the P2P/NotP2P ratio (more P2P records).
2. In addition to the attributes "protocol", "Packet Length", "Src IP", and "Des IP", the accuracy of the classifier depends on the "Time" and "ID" attributes as following:

   2.1- "Arr. Time" plays a significant role in building an accurate classifier

   2.2- "ID" works fine for small P2P/Not-P2P ratio, but does not improve the accuracy for high P2P/Non-P2P ratios.

For a small number of P2P records (<40/60 = 13000), the correlation of the "Src IP" and "Des IP" relationship was lost, so the neural network was struggling to learn the P2P pattern, and the classifier was in need of both the time sequence and successive "ID" to consider in learning the P2P behavior, even though they were not enough; while removing the "ID" harmed the learning process. In any way, for a small number of P2P records, the classifier failed to detect the P2P instances and the use of "Arr. Time" and/or "ID" did not have an important impact.

Above the 50/50 % ratio, when the number of P2P records were equal to 16000, the unison array of the time sequence (inter arrival time) and successive ID in consistence with the packet length, protocol and the {"Src IP", "Des IP"} pair correlation started to establish and the neural network started to learn the P2P behavior (Sensitivity >85%).

With more P2P records (>20000) the neural network classifier started to learn well (Sensitivity > 90%) and the "ID" became useless as the "Arr. Time", "Src IP", and "Des IP" attributes were enough to infer the P2P behavior .

Again, the Correctness measures failed to evaluate the overall trueness of the algorithm. For the P2P/Non-P2P ratios less than 30/70 = 9600 P2P records, the Correctness decreases (from 93.8% to 82%) due to the fact that the Sensitivity was small and P2P records were increasing. For the P2P/Non-P2P ratios between 40/60 and 60/40 the Sensitivity started increasing (from 74.8% to 91.7%) as a result of including a noticeable portion of P2P records in the training set, and the sensitivity started to increase, so the Correctness increased accordingly (form 82.9% to 91.7%). Moreover, for the P2P/Non-P2P ratios of higher than 70/30, where the Sensitivity was high, the Correctness became high too (>96%), considering that Specificity was always high at this interval.

Also, we compared different classifiers of the neural network using the Receiver Operating Characteristic (ROC) curves. The ROC curve depicts the TP rate vs. the FP rate, where the TP rate is indeed the Sensitivity calculated in (2), and the FP rate is:

$$FP\ rate = FP / (FP + TN) = 1 - Specifity \qquad (5)$$

The ROC curve demonstrates two important characteristics: the accuracy of the classifier, and the cost of mistakes (errors) in classifying the inputs.

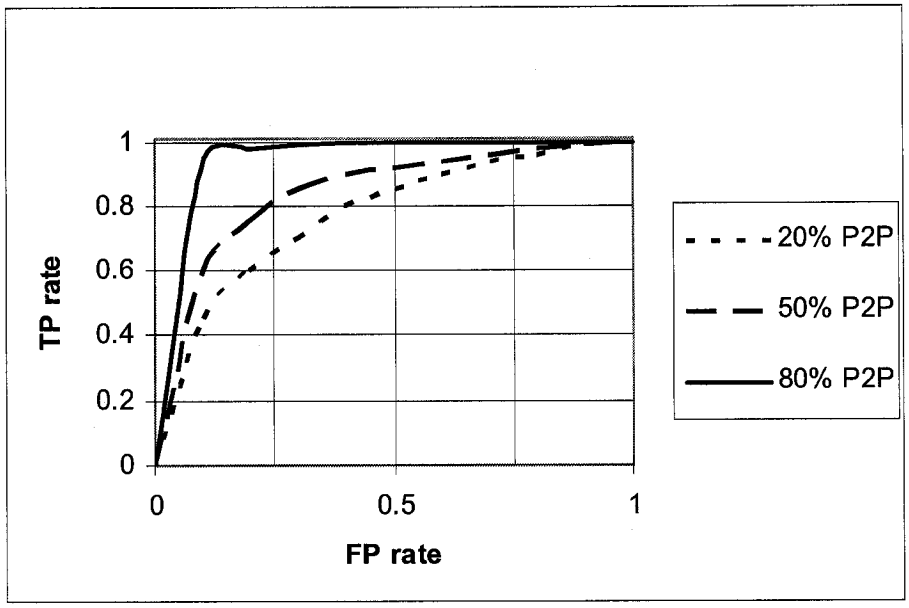


Figure 26: ROC curves for different Mix files

To plot the ROC curves, we select three curves representing three different values of P2P/Non-P2P ratios, Figure 26. One curve represents the dataset in which there are small number of P2P records available (20% = 6400 records), the other curve represent the data set with reasonable number of P2P records (50% = 16000 records), and the third one represents the data set with significant number of P2P records (80% = 25600 records). The closer the ROC curve is to the top-left corner, the more accurate and the less error costly (of wrong classification) the model will be. This is because the top-left corner of the ROC space represents the highest TP rate, (the maximum TP and the minimum FN), and the lowest FP rate (the minimum FP and the maximum TN). The maximum TP and TN means the best accuracy, while the minimum FN and FP means the least cost of error classified inputs. The ROC curve of Figure 26 confirms that as the more number of P2P packets are included in the training set, the classifier becomes more accurate and less costly of erroneous classification.

6.2 Modeling using the Decision Tree

6.2.1 The Decision Tree

Decision Tree is a classification approach that successively divides large heterogeneous data into smaller sets until the most homogeneous sets (classes) are isolated. In the division process, each attribute is compared to a defined value(s) and separated accordingly. Decision tree can be binary where each attribute value has two options only, Figure 27, and the classifier has two classes. Or, it can be N dimension tree where the attribute value is examined against N options, and N classes are resulted, Figure 28.

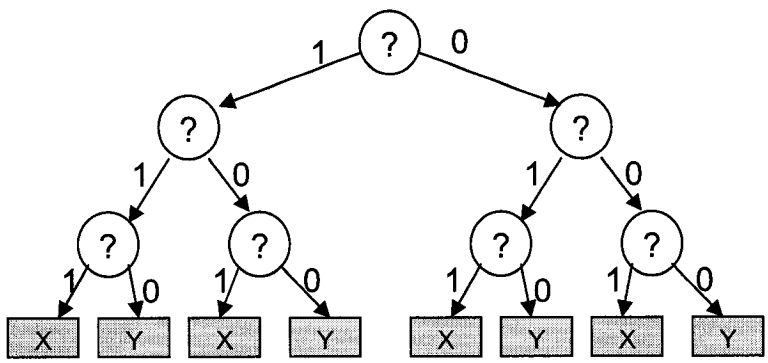


Figure 27: Binary decision tree

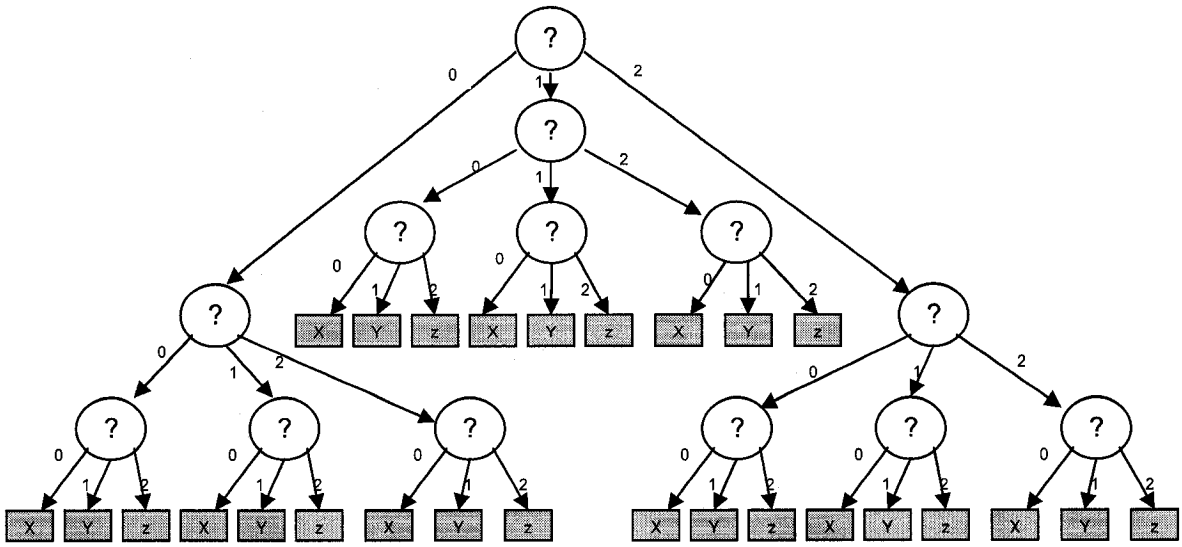


Figure 28: Ternary decision tree

In our study, we used the J48 decision tree; it is an implementation of the C4.5 algorithm [29]. C4.5 is an extension of the basic ID3 algorithm designed by J. Ross Quinlan (1986) to address some issues not dealt with by ID3, such as handling missing data by predicting the missing value based on what is known about the attributes values for the other records, enhancing the pruning process using sub tree replacement and sub tree raising, handling continuous data by dividing them into ranges and improving the splitting mechanism using the Gain Ratio measure instead of the Gain measure itself.

The J48 Decision tree classifier follows the following simple algorithm. It first creates a decision tree based on the attribute values of the available training data. It identifies the attribute that discriminates the various instances most clearly. This feature that is able to tell us most about the data instances so that we can classify them the best is said to have the highest information gain. Now, among the possible values of this feature, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then we terminate

that branch and assign to it the target value that we have obtained. For the other cases, the algorithm looks for another attribute that gives us the highest information gain. Hence, it continue in this manner until it either get a clear decision of what combination of attributes gives us a particular target value, or it run out of attributes. In the event that it runs out of attributes, or if it cannot get an unambiguous result from the available information, it assigns this branch a target value that the majority of the items under this branch possess.

6.2.2 Building the Models

For the modeling phase, we considered the attributes "protocol", "Packet Length", "Src. IP", "Des. IP" and the "Type". We left the "Arrival time" and "ID" attributes because they did not infer any information for the decision tree approach, in contrast, they may mislead the decision tree because, as mentioned, "Arrival time" and "ID" attributes themselves do not infer any information, but the sequencing in arrival time and successive in ID will be useful, the information that the decision tree can not realize but the neural network was able to understand. The selected attributes were divided into two sets: one set with five attributes ("protocol", "Length", "Src. IP", "Des. IP" and "Type"); and another set with three attributes ("protocol", "Length" and "Type"), Table 11.

Table 11: Attribute sets for modeling using J48 decision tree

Attribute Set	attributes
Set # 1	"Protocol", "Length", "Src. IP", "Des. IP" and "Type"
Set # 2	"Protocol", "Length" and "Type"

the J48 was configred as follows:

1- *BinarySplits*: False. This will anbale the tree to split nominal attributes for more than two branches when building the trees.

2- *ConfidenceFactor*: 0.25. The confidence factor used for pruning (smaller values incur more pruning).

3- *Unpruned*: false. To perform pruning process while building the tree.

4- *Train/Test* mode: Fold 5

The six input files of different number of records F1 - F6 (prepared in data pre-processing) were applied to the J48 decision tree using the two attribute sets, J48 decision tree can work with numeric and categorical attributes, so, IP addresses were left nominal without converting them into numeric values. As a result, 12 classifiers were built. Table 12 shows the resulted classified models.

Table 12: The 12 output classifiers using the J48 decision tree, different number of records and two attributes' sets.

File	Attributes' Sets	Time to Build (Sec)	Correctness (%)	TP rate (%) Sensitivity	TN rate (%) Specificity	FP rate (%)
F1 2007 record	Set#1	0.1	84.9	69.8	90.2	9.8
	Set#2	0.03	47	61.1	45.1	54.9
F2 4008 records	Set#1	0.2	89.8	79.6	92.3	7.7
	Set#2	0.09	44.2	63.7	41.9	58.1
F3 8001 records	Set#1	0.3	92.5	85.4	96.6	3.4
	Set#2	0.13	53	62	49.9	50.1
F4 16025 records	Set#1	0.5	93.5	88.8	96.5	3.5
	Set#2	0.59	52.2	68.3	48.8	51.2

F5 32011 records	Set#1	3.3	97.4	97.5	97.4	2.6
	Set#2	1	52.6	73	45.1	54.9
F6 65531 records	Set#1	11	97.6	98.1	97.3	2.7
	Set#2	4.2	55	75.4	46.3	53.7

Figures 29 to 31 illustrate the accuracy measures, the Sensitivity, Specificity and the Correctness measures of Table 12

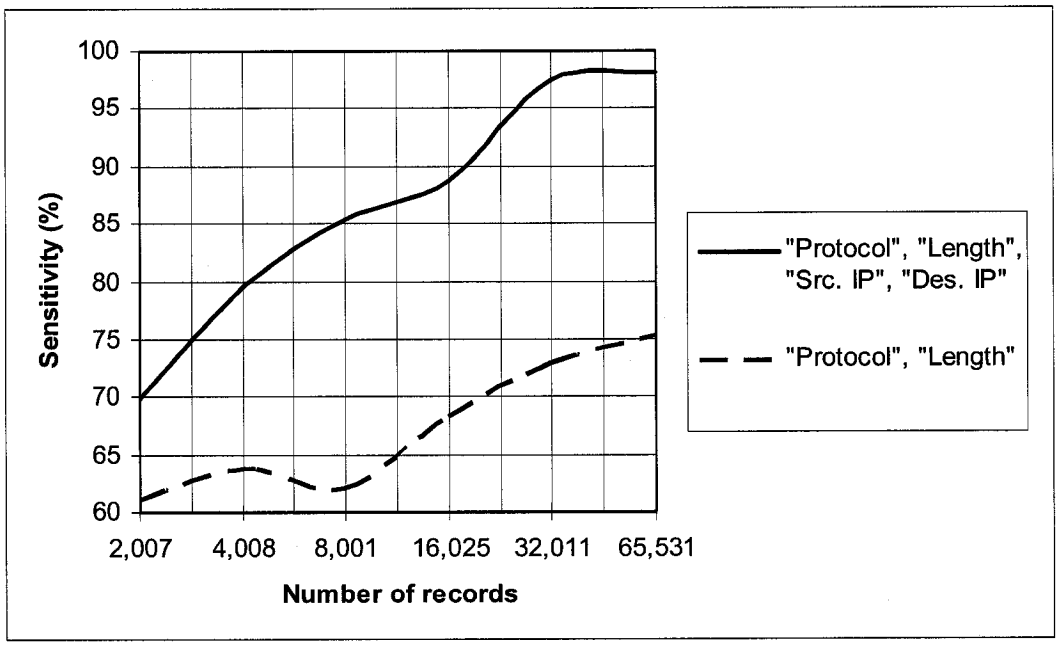


Figure 29: Sensitivity vs. different number of records using two sets of attributes

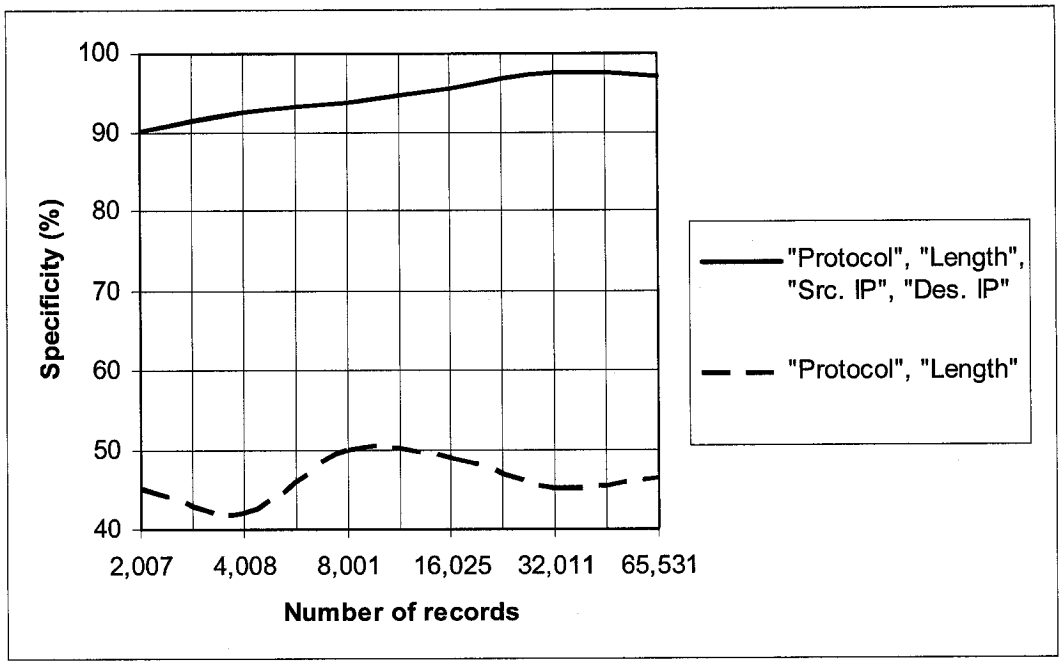


Figure 30: Specificity vs. different number of records using two sets of attributes

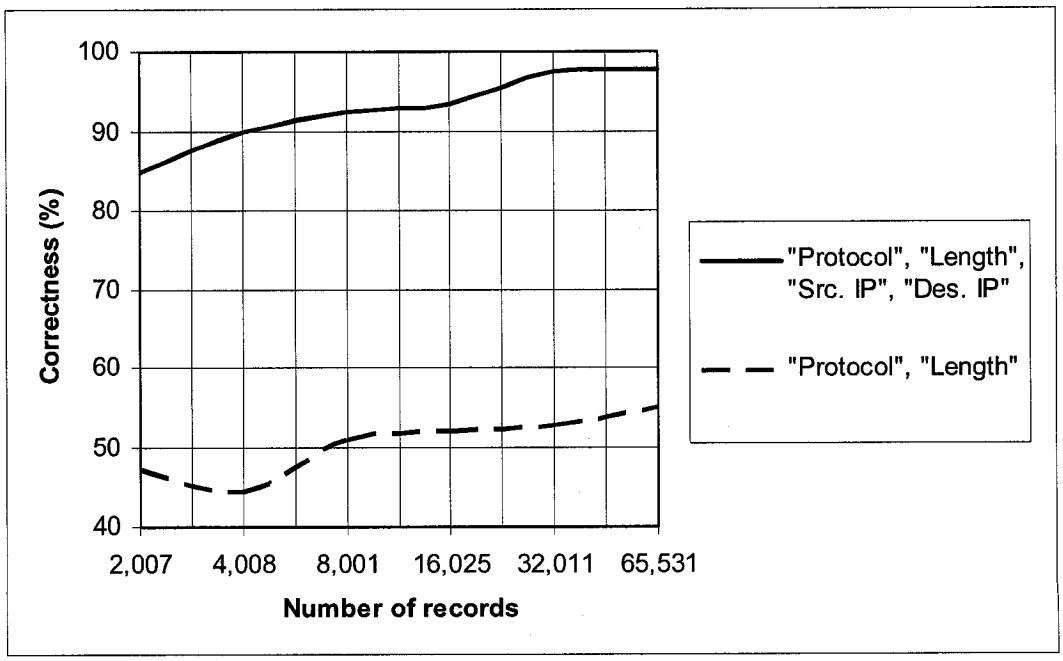


Figure 31: Correctness vs. different number of records using two sets of attributes

6.2.3 Decision Tree Modeling Analysis

According to Table 12 and accuracy measures charts 29 to 31 we conclude that:

1- The models of set #1 have a higher level of accuracy than the models of set #2. In other words, the "protocol" and "length" attributes were few enough for the classifier to learn the P2P behavior, and even to discriminate Non-P2P traffic. However, the use of "Src. IP", "Des. IP" significantly helped the J48 to construct more true P2P and Non-P2P classes.

2- For the models using attribute set #1, as the number of training instances increases the accuracy increases. The best accuracy measures were achieved when the number of records was 32011.

Practically, internet traffic of Non-P2P applications can have both full load packets (1500B) and small load packets (100B) because of different application operations. On the other hand, most P2P applications are characterized by their load packets (1500B). Accordingly, the J48, using only "protocol" and "length", failed to classify the Non-P2P traffic (Specificity < 50%), but it was able to show this common feature of P2P traffic when it reasonably classified 62% - 75% of the P2P traffic. However, the introduction of the "Src. IP" and "Des. IP" attributes enabled the J48 to discriminate the both classes relying on the IP addresses. Also, this is confirmed by the high values of specificity and the increasing values of sensitivity as the number of P2P records increases.

For File 5, when the number of records was 32011, there was a large enough number of P2P packets (10371 records) for J48 to accurately (97.5%) build the P2P tree. Accordingly, we selected the model of set #1 with 32,000 training records to build the decision tree that can be used to classify upcoming traffic.

For this selected model, we used the same Mix files prepared in the pervious section (neural network modeling) to study the response trend of the classifier to different P2P records. Using the same J48 configurations and the two attribute sets, the nine mix files of P2P/Non-P2P traffic were used to build new models. As a result, 9 classifier models were built. Table 13 shows the resulted classification models.

Table 13: The 18 output classifiers using the J48 decision tree, nine different number P2P/Non-P2P ratios of 3200 record file and Set #1 attributes' set.

File	Attributes' Sets	Correctness (%)	Sensitivity (%)	Specificity (%)	FP rate (%)
Mix1	Set#1	96.5	88.1	97.5	2.5
Mix2	Set#1	96.4	90	98	2
Mix3	Set#1	97.1	95.7	97.8	2.2
Mix4	Set#1	97.5	97.2	97.8	2.2
Mix5	Set#1	97.5	97.5	97.5	2.5
Mix6	Set#1	97.7	97.6	97.8	2.3
Mix7	Set#1	96.8	96.5	97.2	2.8
Mix8	Set#1	94	93.2	96.4	3.6
Mix9	Set#1	91.2	90.9	92	8

Figures 32 to 34 illustrate the Correctness, Sensitivity, and Specificity measures of Table-13.

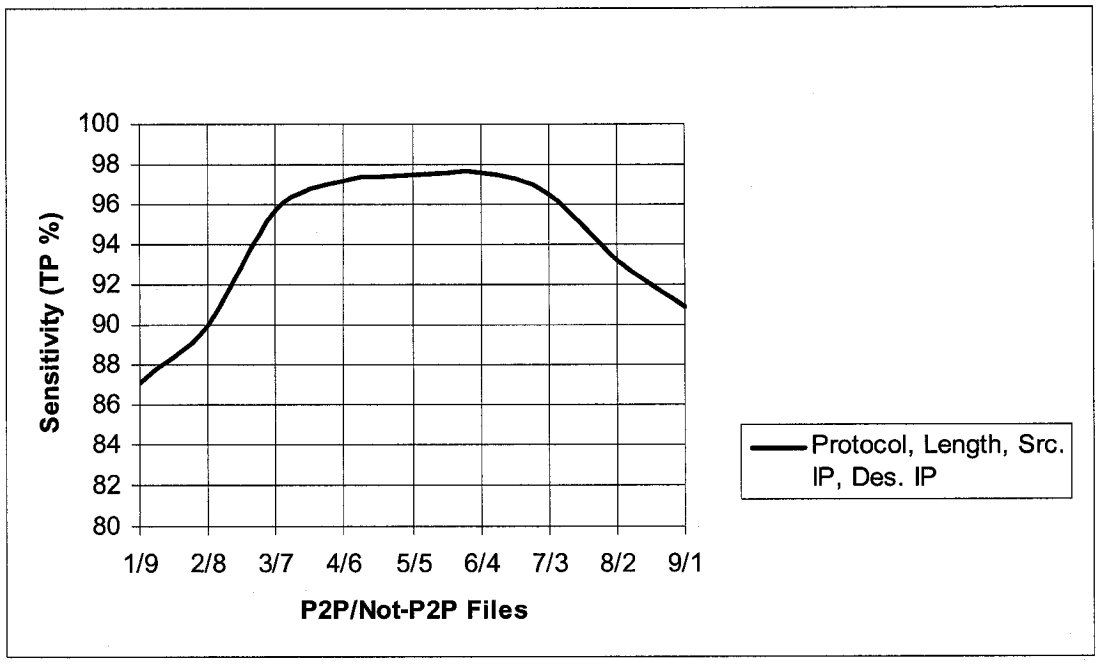


Figure 32: Sensitivity vs. nine different P2P/Non-P2P ratios and Set #1 attributes' set

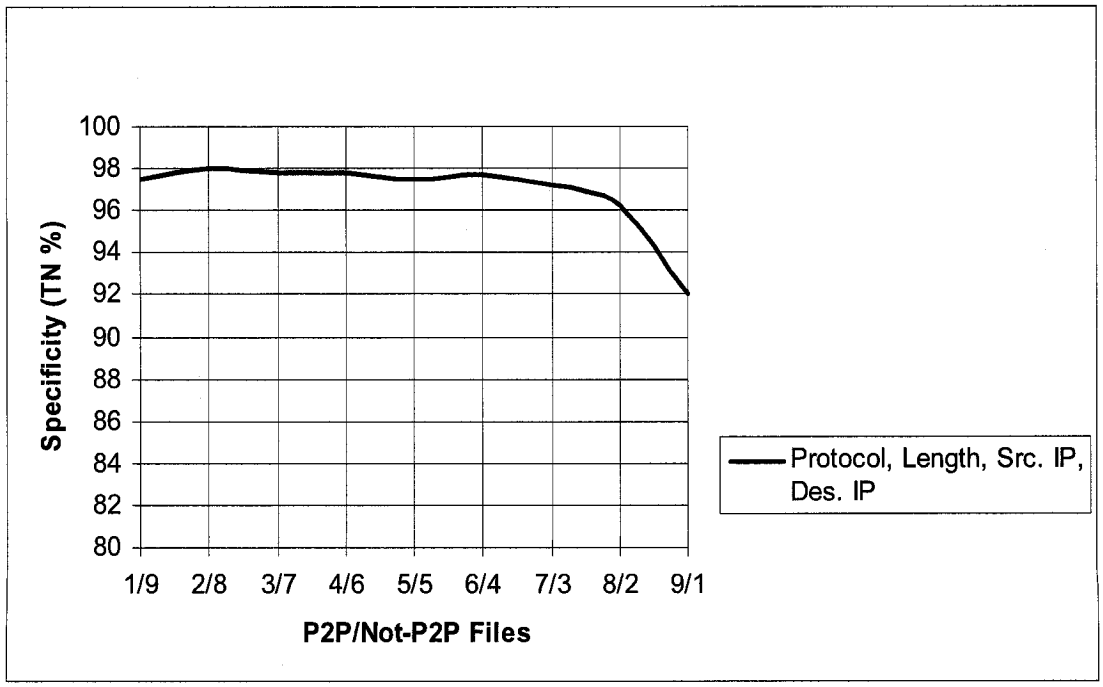


Figure 33: Specificity vs. nine different P2P/Non-P2P ratios and Set #1 attributes' set

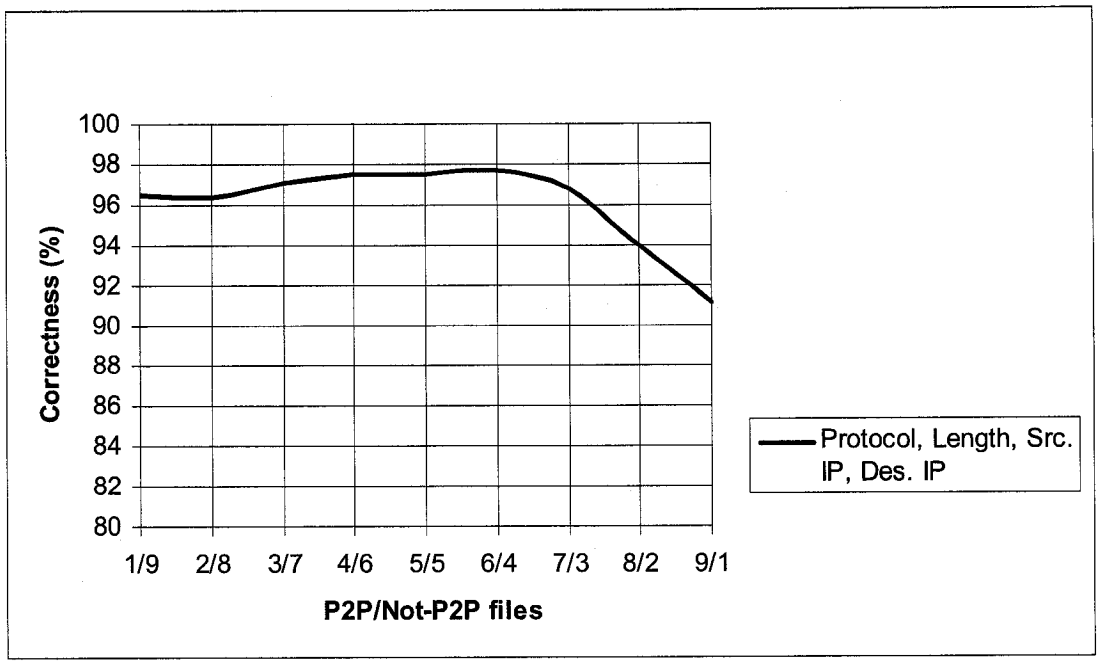


Figure 34: Correctness vs. nine different P2P/Non-P2P ratios and Set #1 attributes' set

From Table 13 and the accuracy measures charts of Mix files we infer that:

1- The classifier started to accurately detect P2P traffic (Sensitivity > 90%) when the P2P/Non-P2P ratio equal to 20/80 %, where the number of P2P records was equal to 6400.

2- When the P2P/Non-P2P goes beyond 70/30 % (the number of P2P packets was equal to 22000) the Sensitivity, and accordingly the Correctness, of the model degrades.

The first observation precisely determined the threshold that J48 can build an accurate classifier, which was confirmed by the J48 classifier built using the six files.

According to the second observation, the retreat of Sensitivity is due to over-fitting phenomena that happens when too many instances are used to train the tree. Practically,

more P2P records implies more P2P packets of the same P2P session, which means more information available about the P2P pattern, and accordingly the over fitting happened when the P2P/Non-P2P ratio was equal to, and greater than 70/30 % (22500 P2P packets). It is noticeable that the specificity did not suffer of this over fitting, and this is because more Non-P2P records imply more distinct Non-P2P application sessions. So, the accuracy of J48 will not be affected, but the number of leaves of Non-P2P classes will increase.

The decision tree relies on the nominal values of the IP addresses to construct the classification tree and accordingly the classes, which, at the first glance, implies that the resulted classifier will not be able to correctly classify new entries that have new IP addresses. In fact, the divide and corner approach of the decision tree indicates that each comparison value depends on the previous resulted node. Therefore, the decision tree can be converted into a set of association rules that construct a chain of relationship connections between attributes.

According to the models performance, we should mention that because different classification models employ different algorithms, the time to build the models and their accuracy should be different also. Figure 37 illustrates the time consumed to build the best model by each technique (neural network and decision tree) using the 32011 training records.

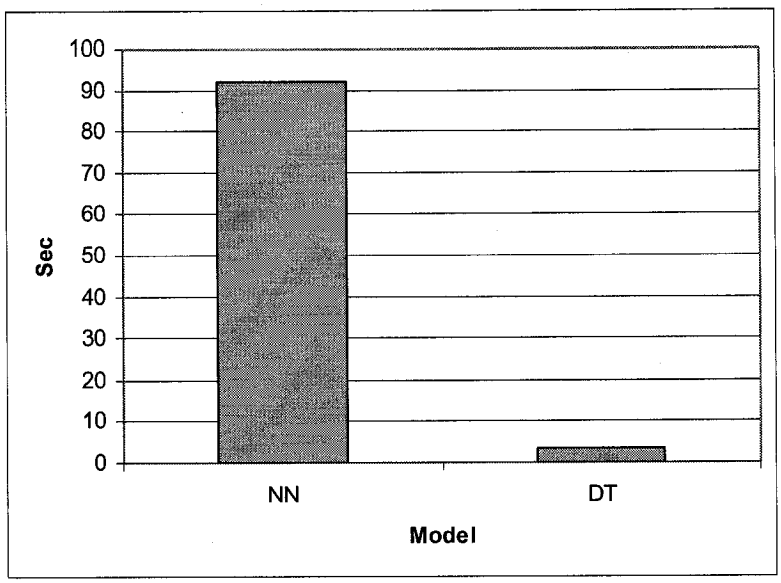


Figure 35 : Time consumed to build the Neural Network (NN) and Decision Tree (DT) models using 32011 training records.

Furthermore, the different techniques required different numbers of P2P training records in order to identify the P2P class correctly. While the neural network required 28800 P2P records to achieve 98.3 % of Sensitivity; the decision tree required 12800 P2P records to achieve 97.5 % sensitivity.

Additionally, both classifiers have a high Specificity measures which indicates a very low false positive rate. Accordingly, the misclassification of Non-P2P traffic is very low. This is important for the ISPs in order not to mistakenly penalize subscribers using non-P2P application.

CHAPTER 7 - CONCLUSIONS

7.1 The contribution of the research:

Throughout this research, we investigated the business impacts of P2P traffic on the Telecom industry and the production and distribution businesses. We calculated an estimated cost 1.45 billion dollars per year to transport P2P traffic across the internet network worldwide. This justifies the network and service providers' interest in finding an effective classification solution for P2P traffic. Also, we elaborated on the other impacts of P2P applications such as the security threats and privacy issues.

We reviewed and discussed many different state-of-the-art researches in P2P networks' evolution, operation and identification solutions. Although the prior solutions offer promising models, they each look at the problem from a different angle, considering various amount of information. Some suffer from limitations such as scalability, speed, and privacy issues.

We demonstrated the application of data mining techniques in P2P traffic classification. Our approach is fast and simple; also, it relies only on the IP layer attributes, eliminating the privacy issues posed by deep packet inspection.

We captured the internet traffic at the gateway of the university. Data was pre-processed and prepared for different data mining requirements; attributes were selected to best describe the P2P patterns. Then data was applied to the MultilayerPerceptron function (a neural network analysis) and J48 decision tree classification techniques. The

proposed classifications relies only on 5 attributes at the IP layer "Arrival time", "Packet "length", "protocol", "source IP address", and "destination IP address". It does not require deep inspection of the packets, and as such, it is easier to deal with the issues of privacy and content security.

Through using neural network and decision tree data mining techniques, we observed that the accuracy of the classification increases significantly when we take into account the source and destination IP addresses. This implies that the accuracy of the classifiers increases when we add information about the peers' connection history; we named this as *community of peers*. Also, it was proved that most P2P applications use large packet size to transfer data files, almost 1500 Bytes.

Throughout this research, we build two classifiers:

1- The neural network classifier using the Multiperceptron function. The classifier showed a very high accuracy (98.3%)of classifying the P2P traffic when using large enough number (>28000) of P2P records and utilizing the "Arrival time", "protocol", "Packet Length", "Source IP", "Destination IP", "Source Port" and "Destination Port".

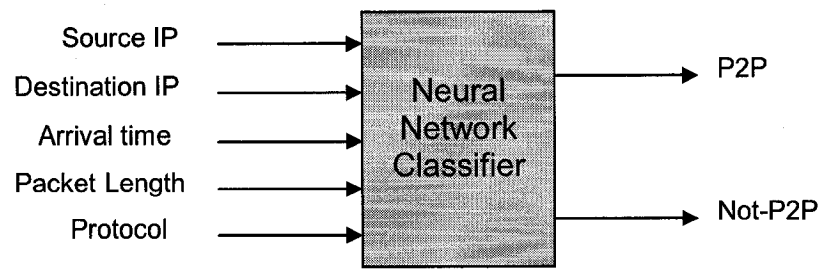


Figure 36: Neural network classifier

2- The decision tree classifier using the J48 (C4.5 algorithm). The classifier showed an excellent accuracy (97.5%) of classifying the P2P traffic when using 13000 of P2P records and utilizing only the "protocol", "Packet Length", "Source IP", "Destination IP", "Source Port" and "Destination Port".

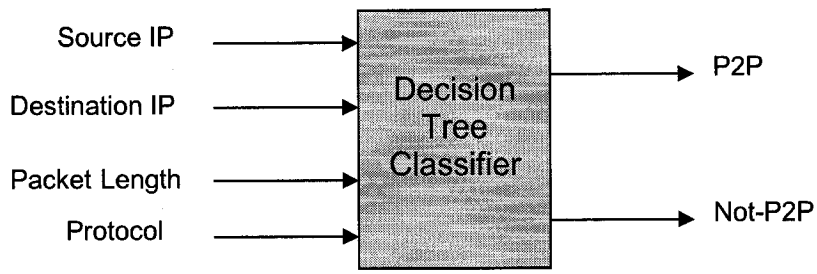


Figure 37: Decision tree classifier

Based on our observations, we make two recommendations:

a. The classifier needs to be implemented within the administrative domain of the individual service provider's networks where each ISP has its pool of IP addresses that they can control.

b. The model needs to be continuously updated to ensure that new communities of peers are detected, and old communities of peers are removed after they stop using peer-to-peer applications. This recommendation is applicable in the light that our proposed solutions require a few seconds to learn and build a new classification model, which can be done as a process continuously running in the background

The different proposed models have different requirements to accurately detect P2P traffic because each technique utilizes a different approach. The neural network required six attributes, 288000 P2P record and 90 seconds in order to detect P2P traffic accurately (98%). The Decision tree was in the middle that required 4 attributes, 4 seconds and around 12000 records to detect 97.5% of P2P traffic.

7.2 Future work

This study was performed on off-line data in order to discover the common characteristics (patterns) of P2P traffic. The next step is to utilize these patterns in investigating stream (online) data mining algorithms that can be instantaneously and efficiently applied to detect P2P traffic. Stream data mining represents an important class of data-intensive applications where packages of information flow dynamically in large volumes, often demanding fast and/or real-time processing of the most current information. Unlike data processing methods for stored datasets, solutions for analyzing streaming data require fast and memory efficient techniques.

Also, we are examining a self-organizing incremental learning neural network algorithm (called ARTMAP) for P2P classification, where both input and output vectors are associated in a learning process, and an internal controller that ensures autonomous system operation in real time.

Our successful approach in detecting P2P traffic can also be considered the base for classifying internet traffic for other purposes, such as intrusion detection, pattern recognition, virus detection, and trend analysis.

7.3 Publications resulted from this research

1- "Classification of Peer-to-Peer Traffic Using Neural Networks", *International Conference on Artificial Intelligence and Pattern Recognition (AIPR-07)*,Orlando, USA, to appear in July 2007.

2- "Peer-to-Peer IP Traffic Classification Using Decision Tree and IP Layer Attributes", *International Journal of Business Data Communications and Networking (IJBDCN)* – accepted for publication in upcoming issue in 2007.

3- "Business Impacts of Peer-to-Peer Traffic on Telecom Networks and Service Providers" Working paper 06-32, University of Ottawa, School of Management, October 2006.

REFERENCES

[1] Steinmetz R. and Wehrle K. "Peer-to-Peer Systems and Applications, Lecture Notes in Computer Science", Springer; 1st edition (October 25, 2005).

[2] Crews K., "*Case Summery, A&M Records, Inc. v. Napster, Inc.*", September 2001

[3] Azzouna, N.B.; Guillemin, F., "Impact of peer-to-peer applications on wide area network traffic: an experimental approach", *IEEE Global Telecommunications Conference, GLOBECOM '04,* Vo. 3, P1544-1548 (29 Nov.-3 Dec. 2004),

[4] Kamei, S.; Kimura, T., "Practicable network design for handling growth in the volume of peer-to-peer traffic", *IEEE Pacific Rim Conference on Communications, Computers*

[5] Subramanian R. and Goodman B., "Peer to Peer Computing: The Evolution of a Disruptive Technology", Idea Group Publishing (February 2005)

[6] CacheLogic®. Research study "Understanding the Impact of P2P: Architecture and Protocols", (2006)

[7] P.Cube Inc. white paper, "Controlling Peer to Peer Bandwidth Consumption", (2003)

[8] Parker A., CacheLogic®. Research study, "P2P in 2005", (27 Nov. 2005)

[9] De Argaez E., "World Broadband Usage in 2005", Computer Industry Almanac. http://www.internetworldstats.com/articles/art130.htm

[10] Norton W., "Internet Service Providers and Peering"
www.equinix.com/pdf/whitepapers/PeeringWP.2.pdf

[11] TeleGeograghy Research, Market research report, " Global Traffic, Bandwidth, and Pricing Trends and wholesale Market Outlook", January 2007

[12] Parker A., CacheLogic®. Research study, "The true Picture of P2P file Sharing",

[13] Steinmetz R. and Wehrle K. "Peer-to-Peer Systems and Applications, Lecture Notes in Computer Science", Springer; 1st edition (October 25, 2005).

[14] CacheLogic®. Technical report "Real-Time Traffic Analysis of File Formats Crossing Peer-to-Peer" (2005)

[15] http://www.eff.org/legal/cases/RIAA_v_Verizon/

[16] Fisher W., "Promises to Keep: Technology, Law, and the Future of Entertainment" Stanford, California, Stanford University Press,pp.340, (2004).

[17] http://www.bailii.org/ew/cases/EWHC/QB/2006/407.html

[18] University of Windsor, Rocking in the Not So Free Virtual World Conference. http://cfl-x.uwindsor.ca/LAW/conference01.htm

[19] Geist M., "Piercing the Peer-to-Peer Myths: An Examination of the Canadian Experience", First Monday, volume 10, number 4 (April 2005)

[20] International Federation of the Phonographic Industry, IFPI, "Legal music downloads triple in 2005; file-sharers take heed of lawsuits", (2005)

[21] Jupiter UK Music Consumer Survey, "Digital Music Landscape Shifting" June, 2005

[22] TruSecure "WildTrends 2003: A look at virus trends in 2003 and a few predictions in 2004" (December 2003).

[23] Lipschutz R. and Clyman J., "*P2P Programs: Popular and Perilous*" http://www.pcpitstop.com/spycheck/p2p.asp

[24] Sen, S., Spatscheck O. and Wang D., "*Accurate, Scalable In-Network Identification of P2P Traffic using Application Signatures*," Proceedings of the 13th International World Wide Web Conference, pp. 512-521, NY, USA, May 2004.

[25] Karagiannis T., Broido A., Faloutsos M. and Klaffy K., "*Transport Layer Identification of P2P Traffic*," Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement (IMC 2004), pp. 121-134, Italy, October 2004.

[26] Karagiannis T., Papagiannaki K. and Faloutsos M., "*BLINC: Multilevel Traffic Classification in the Dark*". Proceedings of ACM SIGCOMM conference, August, 2005. Philadelphia, PA.

[27] Sebastian Zander, Thuy Nguyen, and Grenville Armitage, "*Self-learning IP Traffic Classification based on Statistical Flow Characteristics*" Proc. Passive and Active Measurement workshop PMA Conference, Boston, MA. March 31 - April 01, 2005

[28] Denis Zuev and Andrew W. Moore, "*Traffic Classification using a Statistical Approach*" in the Proceedings of Sixth Passive and Active Measurement Workshop (PAM 2005), March/April 2005, Boston, MA.

[29] Dunham M., "*Data Mining, Introductory and Advanced Topics*", Prentice Hall, (2003).

[30] Maimon O. and Rokach L., "*Data Mining and Knowledge Discovery Handbook* "Springer; 1st (2005).

[31] http://www.tcpdump.org/tcpdump_man.html

[32] http://www.winpcap.org/windump/docs/manual.htm

[33] Kevin Tatroe, Rasmus Lerdorf, Peter MacIntyre "*Programming PHP*", O'Reilly Media; 2nd, (2006).

[34] Ian H. Witten & Frank, E. "*Data Mining, Practical Machine Learning Tool and Techniques*", Elsevier printing, (2005).

[35] Michael J. A. Berry & Gordon S. Linoff, "*Data Mining Techniques for marketing, Sales and Customer Relationship Management*" Wiley Publishing, 2nd, (2004).

APPENDIX

A. The output report of running the neural network using Mix 8 file (80/20 % P2P/Non-P2P) and attributes' set #3 ("Arr. Time", "Protocol", "Length", "Src. IP", "Des. IP" and "Type").

```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -
M 0.2 -N 500 -V 0 -S 0 -E 20 -H i -B -C -I
Relation:     Ready Mix8.txt-
weka.filters.unsupervised.attribute.Remove-R7-8-
weka.filters.unsupervised.attribute.Normalize-
weka.filters.unsupervised.attribute.Remove-R2
Instances:    32000
Attributes:   6
              Time
              Protocol
              Length
              Src IP
              Des IP
              Type
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===

Sigmoid Node 0
    Inputs    Weights
    Threshold    2.3077532374483156
    Node 2    -20.13983050868877
    Node 3    -2.8502499844726135
    Node 4    -13.612658896466288
    Node 5    -2.1075504200533595
    Node 6    -14.640492158860903
Sigmoid Node 1
    Inputs    Weights
    Threshold    -2.3077532351913117
    Node 2    20.139830483745467
    Node 3    2.850249979613095
    Node 4    13.612658364034411
    Node 5    2.107550418560643
    Node 6    14.640492094680278
Sigmoid Node 2
    Inputs    Weights
    Threshold    -12.577706125859212
    Attrib Time    105.2513088102314
    Attrib Protocol    9.930203080368605
    Attrib Length    -0.11987107616347459
    Attrib Src IP    1.5079719102712268
    Attrib Des IP    1.2912396951358356
Sigmoid Node 3
    Inputs    Weights
    Threshold    -10.811405376635948
```

```
    Attrib Time    34.578284787517966
    Attrib Protocol    5.45133994458556
    Attrib Length    -5.33079303198489
    Attrib Src IP    9.734261800035545
    Attrib Des IP    1.7527555061556188
Sigmoid Node 4
    Inputs    Weights
    Threshold    3.086477027893413
    Attrib Time    5.851894684111272
    Attrib Protocol    1.66373391376253
    Attrib Length    -0.23307009112776525
    Attrib Src IP    -57.84842871508041
    Attrib Des IP    1.6972964839753308
Sigmoid Node 5
    Inputs    Weights
    Threshold    23.095291252427156
    Attrib Time    -13.67858653425526
    Attrib Protocol    2.8742758374821684
    Attrib Length    -19.846572820929826
    Attrib Src IP    -39.53128849740574
    Attrib Des IP    0.1839845574675251
Sigmoid Node 6
    Inputs    Weights
    Threshold    1.2425865294108673
    Attrib Time    4.781607933708712
    Attrib Protocol    2.2623327714961285
    Attrib Length    20.202747795724438
    Attrib Src IP    4.788714061192095
    Attrib Des IP    -62.77402738673692
Class NOTP2P
    Input
    Node 0
Class P2P
    Input
    Node 1


Time taken to build model: 92.93 seconds


                === Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances       30559               95.4985 %
Incorrectly Classified Instances       972                4.5015 %
Kappa statistic                          0.8654
Mean absolute error                      0.0637
Root mean squared error                  0.184
Relative absolute error                 19.9148 %
Root relative squared error             46.0034 %
Total Number of Instances            21593
```

=== Detailed Accuracy By Class ===

TP Rate	FP Rate	Precision	Recall	F-Measure	Class
0.948	0.043	0.846	0.948	0.894	NOTP2P
0.957	0.052	0.987	0.957	0.971	P2P

=== Confusion Matrix ===

a	b	<-- classified as
6067	333	a = NOTP2P
1101	24499	b = P2P

B. The output report of running the Decision tree using F1 (2007 records) and attributes' set #1 ("Protocol", "Length", "Src. IP", "Des. IP" and "Type")

```
                    === Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     K2f9.txt-weka.filters.unsupervised.attribute.Remove-R1-
2,7-8
Instances:    2007
Attributes:   5
              Protocol
              Length
              Src IP
              Des IP
              Type
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

Src IP = 137.122.73.74: P2P (6.0/1.0)
Src IP = 207.46.219.62: NOTP2P (46.0)
Src IP = 137.122.67.22: P2P (17.0)
Src IP = 38.99.12.35: NOTP2P (165.0)
Src IP = 85.207.4.137: P2P (1.0)
Src IP = 137.122.71.70: NOTP2P (219.0)
Src IP = 137.122.78.54: NOTP2P (82.0)
Src IP = 137.122.78.21: NOTP2P (31.0)
Src IP = 137.122.76.124: P2P (1.0)
Src IP = 69.242.170.227: P2P (6.0)
Src IP = 137.122.69.177: P2P (7.0)
Src IP = 67.187.60.7: P2P (2.0)
Src IP = 68.35.98.126: P2P (3.0)
Src IP = 137.122.70.23: P2P (17.0)
Src IP = 212.203.28.218: NOTP2P (13.0)
Src IP = 24.57.201.14: P2P (1.0)
Src IP = 137.122.68.243: NOTP2P (33.0)
Src IP = 172.207.103.213: P2P (1.0)
Src IP = 207.226.166.5: NOTP2P (49.0)
Src IP = 216.120.241.17: NOTP2P (11.0)
Src IP = 137.122.76.23: NOTP2P (32.0/1.0)
Src IP = 86.56.11.14: P2P (1.0)
Src IP = 137.122.73.82: P2P (14.0)
Src IP = 81.66.28.51: P2P (1.0)
Src IP = 137.122.73.47: NOTP2P (20.0)
Src IP = 84.4.73.25: P2P (1.0)
Src IP = 137.122.68.81: P2P (2.0)
Src IP = 72.32.11.101: NOTP2P (1.0)
Src IP = 137.122.73.204: NOTP2P (2.0)
Src IP = 67.71.85.56: P2P (1.0)
Src IP = 205.180.86.14: NOTP2P (4.0)
Src IP = 69.235.249.38: P2P (1.0)
```

```
Src IP = 137.122.70.244: P2P (16.0)
Src IP = 66.24.212.145: P2P (1.0)
Src IP = 216.113.194.32: NOTP2P (59.0)
Src IP = 64.74.221.203: NOTP2P (389.0)
Src IP = 137.122.71.189: P2P (5.0)
Src IP = 69.181.176.133: P2P (1.0)
Src IP = 137.122.75.164: NOTP2P (39.0)
Src IP = 18.100.0.67: P2P (2.0)
Src IP = 137.122.66.136
|   Des IP = 83.140.176.214: P2P (0.0)
|   Des IP = 137.122.78.21: P2P (0.0)
|   Des IP = 83.202.46.144: P2P (0.0)
|   Des IP = 137.122.78.54: P2P (0.0)
|   Des IP = 137.122.70.244: P2P (0.0)
|   Des IP = 64.74.221.203: P2P (0.0)
|   Des IP = 38.99.12.35: P2P (0.0)
|   Des IP = 207.46.219.62: P2P (0.0)
|   Des IP = 82.50.88.118: P2P (0.0)
|   Des IP = 137.122.70.23: P2P (0.0)
|   Des IP = 83.157.239.98: P2P (0.0)
|   Des IP = 137.122.74.119: P2P (0.0)
|   Des IP = 137.122.66.136: P2P (0.0)
|   Des IP = 69.242.170.227: P2P (0.0)
|   Des IP = 137.122.68.243: P2P (0.0)
|   Des IP = 137.122.68.142: P2P (0.0)
|   Des IP = 212.203.28.218: P2P (0.0)
|   Des IP = 137.122.77.62: P2P (0.0)
|   Des IP = 137.122.76.23: P2P (0.0)
|   Des IP = 137.122.73.47: P2P (0.0)
|   Des IP = 207.226.166.5: P2P (0.0)
|   Des IP = 137.122.73.82: P2P (0.0)
|   Des IP = 86.56.11.14: P2P (0.0)
|   Des IP = 216.120.241.17: P2P (0.0)
|   Des IP = 81.66.28.51: P2P (0.0)
|   Des IP = 137.122.68.81: P2P (0.0)
|   Des IP = 84.4.73.25: P2P (0.0)
|   Des IP = 207.46.7.1: P2P (0.0)
|   Des IP = 72.32.11.101: P2P (0.0)
|   Des IP = 137.122.69.54: P2P (0.0)
|   Des IP = 137.122.75.42: P2P (0.0)
|   Des IP = 205.180.86.14: P2P (0.0)
|   Des IP = 84.230.149.22: P2P (0.0)
|   Des IP = 137.122.67.212: P2P (0.0)
|   Des IP = 137.122.75.164: P2P (0.0)
|   Des IP = 137.122.71.70: P2P (0.0)
|   Des IP = 86.18.168.45: P2P (0.0)
|   Des IP = 67.77.106.139: P2P (0.0)
|   Des IP = 216.113.194.32: P2P (0.0)
|   Des IP = 137.122.67.230: P2P (0.0)
|   Des IP = 68.35.98.126: P2P (2.0)
|   Des IP = 24.254.82.19: P2P (0.0)
|   Des IP = 70.86.117.98: P2P (0.0)
|   Des IP = 137.122.77.48: P2P (0.0)
|   Des IP = 137.122.67.22: P2P (0.0)
|   Des IP = 132.246.2.7: P2P (0.0)
|   Des IP = 68.253.200.200: P2P (0.0)
|   Des IP = 137.122.76.172: P2P (0.0)
```

```
|    Des IP = 69.233.25.216: P2P (0.0)
|    Des IP = 137.122.75.153: P2P (0.0)
|    Des IP = 137.122.72.190: P2P (0.0)
|    Des IP = 216.52.17.135: P2P (0.0)
|    Des IP = 218.225.247.19: P2P (0.0)
|    Des IP = 209.107.228.91: P2P (0.0)
|    Des IP = 137.122.75.177: P2P (0.0)
|    Des IP = 137.122.75.72: P2P (0.0)
|    Des IP = 72.14.203.99: P2P (0.0)
|    Des IP = 137.122.71.68: P2P (0.0)
|    Des IP = 137.122.73.204: P2P (0.0)
|    Des IP = 80.38.105.85: P2P (0.0)
|    Des IP = 137.122.73.74: P2P (0.0)
|    Des IP = 84.248.155.62: P2P (0.0)
|    Des IP = 219.74.102.25: P2P (0.0)
|    Des IP = 201.143.99.46: P2P (0.0)
|    Des IP = 207.46.7.2: P2P (0.0)
|    Des IP = 67.71.85.56: P2P (0.0)
|    Des IP = 137.122.76.242: P2P (0.0)
|    Des IP = 195.198.188.168: P2P (0.0)
|    Des IP = 86.133.112.19: P2P (0.0)
|    Des IP = 86.20.21.179: P2P (0.0)
|    Des IP = 203.51.55.113: P2P (0.0)
|    Des IP = 67.187.60.7: P2P (0.0)
|    Des IP = 172.207.103.213: P2P (0.0)
|    Des IP = 72.177.238.82: P2P (0.0)
|    Des IP = 212.64.124.99: P2P (0.0)
|    Des IP = 87.123.251.88: P2P (1.0)
|    Des IP = 80.111.194.167: P2P (0.0)
|    Des IP = 137.122.71.189: P2P (0.0)
|    Des IP = 207.46.7.5: P2P (0.0)
|    Des IP = 138.217.77.35: P2P (0.0)
|    Des IP = 137.122.68.93: P2P (0.0)
|    Des IP = 137.122.74.215: P2P (0.0)
|    Des IP = 68.171.165.100: P2P (0.0)
|    Des IP = 137.122.71.75: P2P (0.0)
|    Des IP = 66.176.194.99: P2P (0.0)
|    Des IP = 70.84.96.2: P2P (0.0)
|    Des IP = 151.49.109.109: NOTP2P (12.0)
|    Des IP = 18.100.0.67: P2P (0.0)
|    Des IP = 84.188.142.1: P2P (0.0)
|    Des IP = 81.110.18.99: P2P (0.0)
|    Des IP = 24.57.189.119: P2P (0.0)
|    Des IP = 88.9.52.113: P2P (0.0)
|    Des IP = 81.103.39.206: P2P (0.0)
|    Des IP = 67.181.12.136: P2P (0.0)
|    Des IP = 70.51.81.63: P2P (0.0)
|    Des IP = 81.154.33.244: P2P (0.0)
|    Des IP = 12.120.1.110: P2P (0.0)
|    Des IP = 70.162.206.174: P2P (0.0)
|    Des IP = 70.186.189.205: P2P (0.0)
|    Des IP = 137.122.72.137: P2P (0.0)
|    Des IP = 137.122.74.184: P2P (0.0)
|    Des IP = 70.48.156.140: P2P (0.0)
|    Des IP = 82.59.12.250: P2P (0.0)
|    Des IP = 137.122.74.213: P2P (0.0)
|    Des IP = 137.122.72.108: P2P (0.0)
```

```
|   Des IP = 137.122.69.184: P2P (0.0)
|   Des IP = 71.35.16.237: P2P (0.0)
|   Des IP = 66.230.223.73: P2P (0.0)
|   Des IP = 24.94.218.77: P2P (0.0)
|   Des IP = 81.49.2.212: P2P (0.0)
|   Des IP = 81.71.67.72: P2P (0.0)
|   Des IP = 216.104.162.10: P2P (0.0)
|   Des IP = 84.56.152.120: P2P (0.0)
|   Des IP = 213.196.225.122: P2P (0.0)
|   Des IP = 86.208.128.101: P2P (0.0)
|   Des IP = 221.216.66.173: P2P (0.0)
|   Des IP = 86.213.98.100: P2P (0.0)
|   Des IP = 201.10.168.101: P2P (0.0)
|   Des IP = 81.52.161.76: P2P (0.0)
|   Des IP = 84.29.38.213: P2P (0.0)
|   Des IP = 211.162.1.13: P2P (0.0)
|   Des IP = 83.114.188.164: P2P (0.0)
|   Des IP = 60.228.14.146: P2P (0.0)
|   Des IP = 201.44.209.119: P2P (0.0)
|   Des IP = 24.151.192.62: P2P (0.0)
|   Des IP = 63.227.13.253: P2P (0.0)
|   Des IP = 82.45.184.158: P2P (0.0)
|   Des IP = 65.54.195.185: P2P (0.0)
|   Des IP = 204.11.109.63: P2P (0.0)
|   Des IP = 80.109.52.151: P2P (0.0)
|   Des IP = 137.122.66.109: P2P (0.0)
|   Des IP = 137.122.69.45: P2P (0.0)
|   Des IP = 88.14.149.134: P2P (0.0)
|   Des IP = 137.122.77.54: P2P (0.0)
|   Des IP = 202.5.95.196: P2P (0.0)
|   Des IP = 71.81.231.118: P2P (0.0)
|   Des IP = 219.90.146.114: P2P (0.0)
|   Des IP = 137.122.75.148: P2P (0.0)
|   Des IP = 18.95.7.164: P2P (0.0)
|   Des IP = 82.32.49.99: P2P (0.0)
|   Des IP = 86.212.227.173: P2P (0.0)
|   Des IP = 137.122.72.26: P2P (0.0)
|   Des IP = 68.36.152.34: P2P (0.0)
|   Des IP = 82.37.202.167: P2P (0.0)
|   Des IP = 137.122.74.217: P2P (0.0)
|   Des IP = 161.115.29.9: P2P (0.0)
|   Des IP = 137.122.72.148: P2P (0.0)
|   Des IP = 64.74.223.1: P2P (0.0)
|   Des IP = 137.122.64.64: P2P (0.0)
|   Des IP = 12.215.101.145: P2P (1.0)
|   Des IP = 219.199.104.134: P2P (1.0)
|   Des IP = 70.84.122.106: P2P (0.0)
|   Des IP = 67.15.195.194: P2P (0.0)
|   Des IP = 67.186.118.248: P2P (0.0)
|   Des IP = 137.122.69.145: P2P (0.0)
|   Des IP = 67.189.40.98: P2P (0.0)
|   Des IP = 137.122.70.210: P2P (0.0)
|   Des IP = 69.251.69.65: P2P (0.0)
|   Des IP = 85.166.196.36: P2P (0.0)
|   Des IP = 69.143.15.133: P2P (0.0)
|   Des IP = 137.122.67.75: P2P (0.0)
|   Des IP = 217.165.123.223: P2P (0.0)
```

```
|    Des IP = 172.141.141.119: P2P (0.0)
|    Des IP = 131.252.243.7: P2P (0.0)
|    Des IP = 137.122.65.52: P2P (0.0)
|    Des IP = 81.96.190.128: P2P (0.0)
|    Des IP = 82.41.211.21: P2P (0.0)
|    Des IP = 137.122.69.220: P2P (0.0)
|    Des IP = 24.151.99.151: P2P (0.0)
|    Des IP = 24.65.102.95: P2P (0.0)
|    Des IP = 70.72.183.50: P2P (0.0)
|    Des IP = 87.7.147.72: P2P (2.0)
|    Des IP = 137.122.76.170: P2P (0.0)
|    Des IP = 168.75.214.50: P2P (0.0)
|    Des IP = 71.201.25.5: P2P (0.0)
|    Des IP = 84.102.168.92: P2P (0.0)
|    Des IP = 209.67.78.8: P2P (0.0)
|    Des IP = 69.11.67.124: P2P (1.0)
|    Des IP = 72.228.142.36: P2P (0.0)
|    Des IP = 67.176.76.195: P2P (0.0)
|    Des IP = 70.162.0.10: P2P (0.0)
|    Des IP = 84.104.153.197: P2P (5.0)
|    Des IP = 137.122.73.121: P2P (0.0)
|    Des IP = 24.174.204.91: P2P (0.0)
|    Des IP = 82.72.194.119: P2P (0.0)
|    Des IP = 207.46.7.7: P2P (0.0)
|    Des IP = 68.178.232.99: P2P (0.0)
|    Des IP = 24.1.243.212: P2P (0.0)
|    Des IP = 66.230.182.34: P2P (0.0)
|    Des IP = 137.122.71.19: P2P (0.0)
|    Des IP = 207.46.7.13: P2P (0.0)
|    Des IP = 70.49.10.33: P2P (0.0)
|    Des IP = 81.49.107.218: P2P (0.0)
|    Des IP = 137.122.70.100: P2P (0.0)
|    Des IP = 137.122.67.17: P2P (0.0)
|    Des IP = 84.177.6.59: P2P (0.0)
|    Des IP = 202.156.81.95: P2P (0.0)
|    Des IP = 137.122.74.239: P2P (0.0)
|    Des IP = 62.14.58.70: P2P (0.0)
|    Des IP = 83.83.39.252: P2P (0.0)
|    Des IP = 68.169.177.119: P2P (0.0)
|    Des IP = 137.122.69.197: P2P (0.0)
|    Des IP = 72.29.65.89: P2P (0.0)
|    Des IP = 38.99.208.186: P2P (0.0)
|    Des IP = 137.122.78.49: P2P (0.0)
|    Des IP = 142.59.115.223: P2P (0.0)
|    Des IP = 137.122.66.220: P2P (0.0)
|    Des IP = 137.122.76.251: P2P (0.0)
|    Des IP = 212.225.74.141: P2P (0.0)
|    Des IP = 154.20.94.221: P2P (0.0)
|    Des IP = 60.51.66.127: P2P (0.0)
|    Des IP = 72.255.38.131: P2P (0.0)
|    Des IP = 142.165.165.188: P2P (0.0)
|    Des IP = 86.213.166.65: P2P (0.0)
|    Des IP = 65.117.137.150: P2P (0.0)
|    Des IP = 62.38.69.62: P2P (0.0)
|    Des IP = 66.11.172.69: P2P (0.0)
|    Des IP = 24.2.29.188: P2P (0.0)
|    Des IP = 207.216.16.6: P2P (0.0)
```

```
|    Des IP = 70.108.146.191: P2P (0.0)
|    Des IP = 84.56.70.190: P2P (0.0)
|    Des IP = 69.132.68.251: P2P (0.0)
|    Des IP = 137.122.65.143: P2P (0.0)
|    Des IP = 70.34.198.51: P2P (0.0)
|    Des IP = 80.193.7.140: P2P (0.0)
|    Des IP = 216.80.7.53: P2P (0.0)
|    Des IP = 137.122.68.22: P2P (0.0)
|    Des IP = 69.129.207.101: P2P (0.0)
|    Des IP = 137.122.69.187: P2P (0.0)
|    Des IP = 195.174.208.60: P2P (0.0)
|    Des IP = 24.128.195.252: P2P (0.0)
|    Des IP = 137.122.70.188: P2P (0.0)
|    Des IP = 222.4.54.41: P2P (0.0)
|    Des IP = 66.30.53.83: P2P (0.0)
|    Des IP = 70.125.119.2: P2P (0.0)
|    Des IP = 64.111.210.234: P2P (0.0)
|    Des IP = 12.217.253.238: P2P (0.0)
|    Des IP = 67.68.107.123: P2P (0.0)
|    Des IP = 69.235.213.77: P2P (0.0)
|    Des IP = 24.151.171.129: P2P (0.0)
|    Des IP = 69.250.15.203: P2P (0.0)
|    Des IP = 205.234.230.54: P2P (0.0)
|    Des IP = 81.155.43.190: P2P (0.0)
|    Des IP = 85.200.5.204: P2P (0.0)
|    Des IP = 156.34.210.157: P2P (0.0)
|    Des IP = 67.183.72.32: P2P (0.0)
|    Des IP = 203.206.254.67: P2P (0.0)
|    Des IP = 216.55.169.24: P2P (0.0)
|    Des IP = 200.150.62.140: P2P (0.0)
|    Des IP = 84.174.236.216: P2P (0.0)
|    Des IP = 71.81.202.43: P2P (0.0)
|    Des IP = 137.122.65.7: P2P (0.0)
|    Des IP = 64.236.41.76: P2P (0.0)
|    Des IP = 137.122.76.53: P2P (0.0)
|    Des IP = 70.95.0.230: P2P (0.0)
|    Des IP = 66.65.190.149: P2P (0.0)
|    Des IP = 137.122.74.87: P2P (0.0)
|    Des IP = 217.8.142.204: P2P (0.0)
|    Des IP = 196.202.37.34: P2P (0.0)
|    Des IP = 132.246.2.8: P2P (0.0)
|    Des IP = 137.122.67.116: P2P (0.0)
|    Des IP = 67.87.116.38: P2P (0.0)
|    Des IP = 137.122.66.233: P2P (0.0)
|    Des IP = 68.192.212.247: P2P (0.0)
|    Des IP = 172.203.195.73: P2P (0.0)
|    Des IP = 69.3.182.21: P2P (0.0)
|    Des IP = 203.204.61.18: P2P (0.0)
|    Des IP = 211.36.185.187: P2P (0.0)
|    Des IP = 82.101.158.151: P2P (0.0)
|    Des IP = 207.216.81.150: P2P (0.0)
|    Des IP = 213.244.183.210: P2P (0.0)
|    Des IP = 213.244.183.217: P2P (0.0)
|    Des IP = 64.233.163.89: P2P (0.0)
|    Des IP = 137.122.67.19: P2P (0.0)
|    Des IP = 70.48.113.153: P2P (0.0)
|    Des IP = 70.31.182.122: P2P (0.0)
```

```
|   Des IP = 80.186.139.15: P2P (0.0)
|   Des IP = 213.226.103.55: P2P (2.0)
|   Des IP = 70.229.218.255: P2P (0.0)
|   Des IP = 216.34.88.151: P2P (0.0)
|   Des IP = 216.34.32.118: P2P (0.0)
|   Des IP = 65.97.28.116: P2P (0.0)
|   Des IP = 72.145.123.15: P2P (0.0)
|   Des IP = 137.122.78.5: P2P (0.0)
|   Des IP = 137.122.75.212: P2P (0.0)
|   Des IP = 84.251.145.148: P2P (0.0)
|   Des IP = 69.196.128.122: P2P (0.0)
|   Des IP = 12.226.42.214: P2P (0.0)
|   Des IP = 67.185.35.236: P2P (0.0)
|   Des IP = 219.128.24.43: P2P (0.0)
|   Des IP = 137.122.69.177: P2P (0.0)
|   Des IP = 82.60.190.33: P2P (0.0)
|   Des IP = 70.28.215.115: P2P (0.0)
|   Des IP = 82.161.48.3: P2P (0.0)
|   Des IP = 86.192.65.251: P2P (0.0)
|   Des IP = 24.28.255.16: P2P (0.0)
Src IP = 137.122.69.250: P2P (2.0)
Src IP = 128.208.35.146: P2P (1.0)
Src IP = 83.202.46.144: P2P (4.0)
Src IP = 137.122.72.190: NOTP2P (37.0)
Src IP = 137.122.75.148: P2P (8.0)
Src IP = 69.233.25.216: P2P (1.0)
Src IP = 137.122.76.172: P2P (5.0)
Src IP = 218.225.247.19: P2P (1.0)
Src IP = 132.246.2.7: NOTP2P (28.0)
Src IP = 209.107.228.91: P2P (2.0)
Src IP = 137.122.75.153: P2P (4.0)
Src IP = 71.35.16.237: P2P (1.0)
Src IP = 24.94.218.77: P2P (1.0)
Src IP = 80.38.105.85: P2P (2.0)
Src IP = 207.46.7.1: NOTP2P (1.0)
Src IP = 137.122.71.68: P2P (4.0)
Src IP = 24.57.189.119: P2P (1.0)
Src IP = 70.86.117.98: NOTP2P (8.0)
Src IP = 86.20.21.179: P2P (1.0)
Src IP = 137.122.68.93: NOTP2P (2.0)
Src IP = 137.122.69.54: P2P (5.0)
Src IP = 72.14.203.99: NOTP2P (4.0)
Src IP = 24.78.38.132: P2P (1.0)
Src IP = 195.198.188.168: P2P (4.0)
Src IP = 86.133.112.19: P2P (1.0)
Src IP = 137.122.75.42: P2P (21.0)
Src IP = 137.122.76.242: P2P (2.0)
Src IP = 137.122.74.119: P2P (7.0)
Src IP = 216.52.17.135: NOTP2P (6.0)
Src IP = 137.122.77.62: P2P (3.0)
Src IP = 137.122.72.148: P2P (2.0)
Src IP = 80.111.194.167: P2P (1.0)
Src IP = 151.49.109.109: NOTP2P (7.0)
Src IP = 67.77.106.139: P2P (3.0)
Src IP = 142.179.159.139: P2P (1.0)
Src IP = 137.122.69.184: NOTP2P (2.0)
Src IP = 80.193.7.140: P2P (2.0)
```

```
Src IP = 207.46.7.2: NOTP2P (1.0)
Src IP = 24.180.88.243: P2P (1.0)
Src IP = 137.122.74.217: P2P (3.0)
Src IP = 81.110.18.99: P2P (2.0)
Src IP = 137.122.73.252: P2P (1.0)
Src IP = 137.122.67.230: P2P (5.0)
Src IP = 85.166.196.36: P2P (1.0)
Src IP = 137.122.71.75: P2P (5.0)
Src IP = 137.122.78.49: P2P (10.0)
Src IP = 18.95.7.164: P2P (7.0)
Src IP = 137.122.68.86: NOTP2P (1.0)
Src IP = 63.227.13.253: P2P (1.0)
Src IP = 137.122.68.142: P2P (1.0)
Src IP = 24.28.255.16: P2P (2.0)
Src IP = 172.146.39.237: P2P (2.0)
Src IP = 69.143.15.133: P2P (1.0)
Src IP = 213.113.216.85: P2P (1.0)
Src IP = 207.46.7.5: NOTP2P (1.0)
Src IP = 70.84.96.2: NOTP2P (1.0)
Src IP = 137.122.75.177: P2P (1.0)
Src IP = 137.122.68.140: NOTP2P (1.0)
Src IP = 137.122.75.72: P2P (2.0)
Src IP = 137.122.75.212: P2P (17.0)
Src IP = 137.122.64.64: NOTP2P (4.0)
Src IP = 137.122.70.210: NOTP2P (6.0)
Src IP = 68.125.51.210: P2P (3.0)
Src IP = 88.14.149.134: P2P (3.0)
Src IP = 69.251.69.65: P2P (1.0)
Src IP = 137.122.66.109: P2P (2.0)
Src IP = 82.41.211.21: P2P (1.0)
Src IP = 202.5.95.196: NOTP2P (3.0)
Src IP = 137.122.67.212
|   Length <= 44: NOTP2P (3.0/1.0)
|   Length > 44: P2P (3.0)
Src IP = 219.90.146.114: P2P (2.0)
Src IP = 217.165.123.223: P2P (2.0)
Src IP = 68.253.200.200: P2P (4.0)
Src IP = 86.212.227.173: P2P (1.0)
Src IP = 68.36.152.34: P2P (1.0)
Src IP = 82.37.202.167: P2P (1.0)
Src IP = 137.122.72.26: P2P (12.0)
Src IP = 161.115.29.9: P2P (1.0)
Src IP = 72.177.238.82: P2P (1.0)
Src IP = 137.122.77.37: NOTP2P (1.0)
Src IP = 216.104.162.10: NOTP2P (1.0)
Src IP = 67.15.195.194: NOTP2P (2.0)
Src IP = 204.11.109.63: NOTP2P (5.0)
Src IP = 84.231.153.135: P2P (1.0)
Src IP = 67.186.118.248: P2P (1.0)
Src IP = 70.162.0.10: P2P (1.0)
Src IP = 65.54.195.185: NOTP2P (3.0)
Src IP = 70.84.122.106: NOTP2P (8.0)
Src IP = 137.122.69.45: P2P (1.0)
Src IP = 137.122.74.213: P2P (1.0)
Src IP = 70.49.10.33: P2P (1.0)
Src IP = 172.141.141.119: P2P (2.0)
Src IP = 87.7.147.72: P2P (3.0)
```

```
Src IP = 81.96.190.128: P2P (1.0)
Src IP = 137.122.65.52: P2P (3.0)
Src IP = 68.125.51.9: P2P (1.0)
Src IP = 206.116.129.68: P2P (1.0)
Src IP = 137.122.77.54: P2P (1.0)
Src IP = 202.156.81.95: P2P (1.0)
Src IP = 84.230.149.22: P2P (1.0)
Src IP = 24.151.99.151: P2P (1.0)
Src IP = 137.122.74.218: P2P (1.0)
Src IP = 70.137.145.71: P2P (1.0)
Src IP = 168.75.214.50: NOTP2P (3.0)
Src IP = 137.122.76.170: NOTP2P (2.0)
Src IP = 69.11.67.124: P2P (3.0)
Src IP = 71.201.25.5: P2P (1.0)
Src IP = 67.176.76.195: P2P (4.0)
Src IP = 72.228.142.36: P2P (1.0)
Src IP = 137.122.72.108: P2P (4.0)
Src IP = 137.122.69.145: P2P (3.0)
Src IP = 84.104.153.197: P2P (8.0)
Src IP = 24.174.204.91: P2P (3.0)
Src IP = 24.1.243.212: P2P (1.0)
Src IP = 137.122.71.115: P2P (1.0)
Src IP = 137.122.67.17: NOTP2P (2.0)
Src IP = 209.67.78.8: NOTP2P (8.0)
Src IP = 137.122.70.100: NOTP2P (15.0)
Src IP = 137.122.73.121: P2P (1.0)
Src IP = 137.122.71.19: NOTP2P (5.0)
Src IP = 66.230.182.34: NOTP2P (3.0)
Src IP = 137.122.66.220: NOTP2P (2.0)
Src IP = 137.122.67.75: P2P (1.0)
Src IP = 137.122.72.17: P2P (1.0)
Src IP = 83.83.39.252: P2P (1.0)
Src IP = 38.101.109.38: P2P (1.0)
Src IP = 68.178.232.99: NOTP2P (20.0)
Src IP = 69.132.68.251: P2P (1.0)
Src IP = 84.181.10.94: P2P (1.0)
Src IP = 207.46.7.7: NOTP2P (1.0)
Src IP = 142.59.115.223: P2P (1.0)
Src IP = 137.122.69.220: P2P (1.0)
Src IP = 62.14.58.70: P2P (1.0)
Src IP = 137.122.74.239: P2P (2.0)
Src IP = 66.11.172.69: P2P (1.0)
Src IP = 68.169.177.119: P2P (2.0)
Src IP = 38.99.208.186: NOTP2P (2.0)
Src IP = 137.122.65.143: NOTP2P (54.0/1.0)
Src IP = 137.122.69.197: NOTP2P (1.0)
Src IP = 154.20.94.221: P2P (7.0)
Src IP = 207.46.7.13: NOTP2P (1.0)
Src IP = 12.217.253.238: P2P (1.0)
Src IP = 212.225.74.141: P2P (2.0)
Src IP = 60.51.66.127: P2P (1.0)
Src IP = 137.122.72.137: P2P (2.0)
Src IP = 72.29.65.89: NOTP2P (17.0)
Src IP = 137.122.70.188: NOTP2P (6.0)
Src IP = 69.129.207.101: NOTP2P (3.0)
Src IP = 137.122.68.22: NOTP2P (3.0)
Src IP = 24.89.19.172: P2P (1.0)
```

```
Src IP = 137.122.67.116: NOTP2P (4.0/1.0)
Src IP = 137.122.72.50: P2P (1.0)
Src IP = 216.80.7.53: NOTP2P (6.0)
Src IP = 85.200.5.204: P2P (1.0)
Src IP = 137.122.71.253: P2P (1.0)
Src IP = 137.122.76.251: P2P (1.0)
Src IP = 67.68.107.123: P2P (1.0)
Src IP = 81.155.43.190: P2P (6.0)
Src IP = 69.235.213.77: P2P (1.0)
Src IP = 64.111.210.234: NOTP2P (2.0)
Src IP = 24.65.102.95: P2P (1.0)
Src IP = 137.122.70.243: NOTP2P (1.0)
Src IP = 70.125.119.2: P2P (1.0)
Src IP = 156.34.210.157: P2P (1.0)
Src IP = 69.250.15.203: P2P (2.0)
Src IP = 70.226.155.65: P2P (1.0)
Src IP = 205.234.230.54: NOTP2P (17.0)
Src IP = 80.42.233.3: P2P (2.0)
Src IP = 69.3.182.21: P2P (1.0)
Src IP = 71.81.202.43: P2P (1.0)
Src IP = 207.216.81.150: P2P (1.0)
Src IP = 70.229.218.255: P2P (2.0)
Src IP = 70.49.222.26: P2P (1.0)
Src IP = 66.65.190.149: P2P (2.0)
Src IP = 24.17.255.0: P2P (1.0)
Src IP = 70.31.182.122: P2P (1.0)
Src IP = 216.55.169.24: NOTP2P (5.0)
Src IP = 64.236.41.76: NOTP2P (1.0)
Src IP = 67.183.72.32: P2P (1.0)
Src IP = 137.122.67.19: NOTP2P (15.0)
Src IP = 132.246.2.8: NOTP2P (2.0)
Src IP = 221.241.191.52: P2P (1.0)
Src IP = 83.226.183.234: P2P (1.0)
Src IP = 24.17.163.243: P2P (1.0)
Src IP = 211.36.185.187: P2P (3.0)
Src IP = 68.192.212.247: P2P (1.0)
Src IP = 172.203.195.73: P2P (1.0)
Src IP = 137.122.66.233: P2P (1.0)
Src IP = 137.122.74.87: P2P (2.0)
Src IP = 69.196.128.122: P2P (1.0)
Src IP = 137.122.65.7: P2P (1.0)
Src IP = 64.233.163.89: NOTP2P (2.0)
Src IP = 70.28.215.115: P2P (1.0)
Src IP = 142.104.217.169: P2P (2.0)
Src IP = 219.74.102.25: P2P (1.0)
Src IP = 80.186.139.15: P2P (1.0)
Src IP = 213.226.103.55: P2P (2.0)
Src IP = 86.192.65.251: P2P (2.0)
Src IP = 137.122.76.53: P2P (1.0)
Src IP = 137.122.78.57: P2P (1.0)
Src IP = 219.128.24.43: P2P (2.0)
Src IP = 213.196.225.122: P2P (1.0)
Src IP = 213.244.183.210: NOTP2P (1.0)
Src IP = 213.244.183.217: NOTP2P (1.0)
Src IP = 137.122.78.5: P2P (1.0)
Src IP = 82.161.48.3: P2P (2.0)
Src IP = 216.34.88.151: NOTP2P (1.0)
```

```
Src IP = 69.231.89.187: P2P (1.0)

Number of Leaves  :     544

Size of the tree :      547


Time taken to build model: 0.12 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1704               84.9028 %
Incorrectly Classified Instances       303               15.0971 %
Kappa statistic                          0.777
Mean absolute error                      0.0583
Root mean squared error                  0.207
Relative absolute error                 16.4564 %
Root relative squared error             49.2071 %
Total Number of Instances             2007

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.902     0.302       0.917     0.902     0.906   NOTP2P
  0.698     0.151       0.991     0.698     0.819   P2P
```

C. The output report of running the Decision tree using Mix6 (60/40% P2P/Non-P2P) and attributes' set #3.

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     Mix6.txt-weka.filters.unsupervised.attribute.Remove-R5-6
Instances:    32000
Attributes:   5
              Protocol
              Length
              Src IP
              Des IP
              Type
Test mode:    5-fold cross-validation

=== Classifier model (full training set) ===


Number of Leaves  :     12245

Size of the tree :      12251


Time taken to build model: 93.54 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       31260               97.6906 %
Incorrectly Classified Instances       740                2.3094 %
Kappa statistic                          0.9794
Mean absolute error                      0.0157
Root mean squared error                  0.0848
Relative absolute error                  3.2643 %
Root relative squared error             17.3144 %
Total Number of Instances            32000

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.978     0.024      0.979     0.978     0.974    NOTP2P
  0.976     0.022      0.972     0.976     0.975    P2P


=== Confusion Matrix ===

     a     b   <-- classified as
 12522   278 |     a = NOTP2P
   461 18739 |     b = P2P
```