VoIP Traffic Classification in IPSec Tunnels

Taner Yildirim
Electrical & Computer Engineering
RMIT University
Melbourne, Australia
taner.yildirim@rmit.edu.au

Dr. PJ Radcliffe
Electrical & Computer Engineering
RMIT University
Melbourne, Australia
pjr@rmit.edu.au

***Abstract*— Research in traffic classification has become more challenging with the emergence of new applications and new ways to hide the true nature of traffic. The accuracy of traffic identification methods has also become more important due to the greater use of delay sensitive applications such as VoIP and video over IP which need to be identified and given priority. Traditional techniques such as header and payload inspection are not providing sufficient information to identify traffic types due to the usage of non-standard ports, tunnelling and encryption. Promising methods have been proposed based around the statistical behaviour of traffic flow. Although these methods can achieve quite high accuracies in non-encrypted traffic flows, traffic identification of encrypted traffic flows is still in its early stages. In this paper, we will review the recent work done on encrypted traffic identification, particularly network layer encryption using statistical techniques and propose a remarkably simple technique for VoIP traffic identification in IPSec peer to peer tunnels. More importantly it is shown that VoIP/non-VoIP classification can be used to dramatically improve VoIP QoS and may be used to effectively block non-VoIP traffic in an IPSec tunnel. These results point to the usefulness of the technique and the desirability to find more discriminating VoIP identification algorithms for IPSec tunnels.**

Keywords—VoIP; IPSec; Tunnelled Traffic; Machine Learning

I. INTRODUCTION

Accurate traffic classification is quite critical for network management in enterprise or campus networks. Management of these networks requires complex checking of every packet to ensure network security and conformance with institutional policies. For instance in an enterprise network, file sharing by means of torrent networks is often banned due to its complicated copyright issues and bandwidth costs, online gaming is banned, and accessing certain web sites is blocked due to their illegality. There are a variety of network management techniques available such as using proxy servers and/or statefull firewalls but these often rely on accurate application identification. New threats to accurate identification mean that blockage will be far from ideal in tomorrow's networks.

Application identification can be thwarted by quite simple techniques such as tunneling or encrypting desired applications. For example an increasingly common technique used by students is to use a peer to peer tunnel to hide Bittorrent downloads. Up to date network traffic identification in tunneled scenarios is quite immature and this field needs more research for very practical reasons - if encryption and tunneling become wide spread then firewalls and proxies become far less effective compromising both network security and bandwidth control. Even without tunneling or encryption, traditional techniques such as port blocking or deep packet inspection can be bypassed quite easily. Port altering is quite common in bandwidth hungry applications such as Bittorent. Usually certain ports are not blocked by firewalls or proxy devices, such as HTTP port 80. However, emerging Bittorent client software can send its traffic through port 80. In this case bandwidth usage can increase dramatically and time critical traffic such as VoIP can be badly affected.

While security and access control is a critical, modern networks also have to achieve and acceptable Quality of Service (QoS). For instance a good VoIP implementation requires that the delay, jitter and bit error rate of the link should be minimized. The delay introduced by traffic classification can reduce the QoS thus classification must be both accurate and fast.

This paper examines the specific and important case of VoIP/non-VoIP identification in a peer to peer, network layer encrypted tunnels. This has practical importance as secure voice may well be allowed by corporate policies, but it is currently impossible to stop the tunnel being used for non-VoIP protocols such as bit torrent downloads. Experimental results confirm that VoIP identification in an IPSec tunnel is possible and can be used to keep VoIP QoS acceptable even under extreme network conditions.

The rest of this paper organized as follows. Section II reviews the relevant research for traffic classification using Machine Learning Algorithms and Encrypted Application Classification techniques. In both cases, achieved accuracies are discussed. Section III outlines an algorithm that enables VoIP classification within an IPSec tunnel. Section IV discusses results and accuracy of the method proposed. Section V concludes the paper and considers some further implications of identifying encrypted traffic and future research directions.

II. CURRENT WORK

The literature review section of this paper is divided into two major parts. In the first part, the usefulness of VoIP identification in tunneled networks is discussed. In the second part, the recent research efforts on traffic identification are discussed.

A. *The Benefits of VoIP Identification in Tunnels to Network Administration*

In this section the usefulness of VoIP identification in tunneled networks will be discussed and its benefit to network administration explained. The benefits justify the effort of trying to identify VoIP in an IPSec tunnel. VoIP implementation is in great demand for modern networks as it is much cheaper than traditional telephony.

To understand why it is needed to identify VoIP traffic in secured networks consider Figure 1, a common configuration for a corporate network.

1) *Proxy Servers and affect on Network*

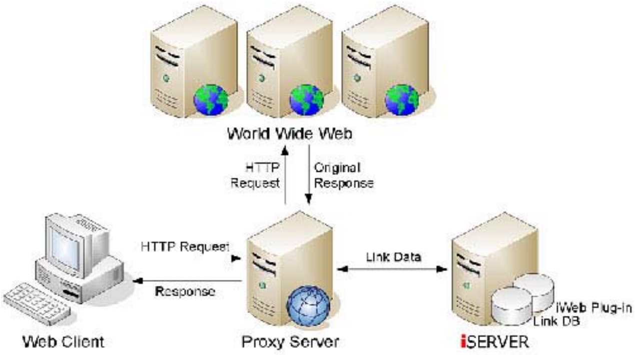

Figure 1. Using proxy server in Networks

A proxy server acts as an intermediate device between a client and a destination, usually a server. The advantage of a proxy server is that it can enforce enterprise policies, enhance security and cache resources locally and so reduce external traffic.

Although proxy servers increase the security levels by protecting internal hosts from unknown or unwanted communications they can also be seen as a bottleneck which can introduce delays and jitter. Therefore, when a VoIP session is tunneled through a proxy device, identifying VoIP and giving appropriate priority in a timely manner is critical to maintaining QoS.

2) *Bandwidth Limitations*

Limited bandwidth availability is a major concern for VoIP traffic. While VoIP adds little to network traffic it can be badly delayed when other protocols make heavy network demands, for example a large FTP session. With un-encrypted traffic, traditional packet inspection can give priority to VoIP and slow down FTP and so maintain an acceptable VoIP QoS. When VoIP is hidden in an encrypted tunnel it cannot be identified and given priority thus the QoS becomes unacceptable.

3) *Banned Tunneles Due to Corporate Policies*

Another scenario is where enterprise policy dictates that all tunneled traffic is banned to ensure a proxy server can properly function. If tunneled traffic is allowed then it becomes quite trivial to initiate a Bittorent session using a tunnel and so bypass the proxy. Since Bittorent is a major concern in terms of both network bandwidth and copyright laws many enterprises would like to stop tunneling but cannot do so because of the wide spread and legitimate use of peer to peer tunnels for VoIP. If it is possible to identify VoIP (and other legitimate traffic) within a tunnel then a proxy server could successfully allow legitimate tunnels and block illegitimate tunnels.

B. *Machine Learning for IP Traffic Classification*

The previous section discussed why it is important to identify VoIP in an IPSec tunnel. This section examines existing research on general classification which may be useful for our goal. The vast majority of relevant papers come from the field of Machine Learning. The concept of Machine Learning is considered to be in the field of artificial intelligence, and hence it can be defined as a computing process of acquiring new knowledge depending on existing knowledge. As discussed in [1] it is a process of *structural pattern recognition* from given data. In the first appearance of Machine Learning technique in classification were used for intrusion detection [2]. For the purpose of this paper we use three sub-fields of Machine Learning Algorithms; these are Supervised Learning methods, Clustering Learning methods (Unsupervised) and Hybrid Learning approaches.

1) *Supervised Learning Algorithms*

Supervised algorithms are type of algorithms which are fed with a training set (i.e. a data set in which there exists a pair of data consisting of an input and a desired output). The efficiency of supervised algorithms very much depends on the provided training set which is used to classify the real data. As long as the network traffic in question is to be considered homogeneous (i.e. does not change its behavior dramatically in time) this type of method can be used.

A promising method was provided by Crotti et al [3], where supervised learning has been employed with the three extractable properties of IP packets; packet length, inter-arrival time and order of arrival. They used so called protocol fingerprints which are identified from PDF vectors. After constructing the PDF vectors, they used a score based algorithm from 0 to 1 in order to probabilistically estimate which incoming flow can be classified for a given PDF vector. Results show that with 91% accuracy they can identify HTTP, SMTP and POP3 protocols. Although promising, they have not employed algorithm to delay sensitive applications such as VoIP. Also time it takes for their algorithm to identify packets has not been discussed. For applications like VoIP packet loss, jitter and delay is quite critical. Therefore overall processing time of the algorithm is a critical measure for efficiency. They also used HTTP as an application but this is problematic today as many new applications hide themselves by masquerading as HTTP. It is also assumed that the initial point of the flows can be caught. Such capture may not be possible when the traffic is tunneled.

A supervised algorithm employed by Li [4] uses the C4.5 decision algorithm [5]. In their paper Internet flow is categorized using distinct classes such as web-browsing, mail, bulk ftp, attack, peer-to-peer, database, service, and interactive. They results showed about 99% accuracy using 1875 flows as a training set to feed C4.5. They also pointed out the issue of accuracy versus complexity in their work. Although tunneled traffic mentioned there is no application of their algorithm to VPN or tunneled traffic in their research.

A paper by Moore [6] used Naïve Bayes technique to classify different application types. They used hand-classified data to use as a training set. They also gathered various feature sets from their training set such as flow length, port number and inter-arrival time. However, the proposed technique cannot be applied to tunneled traffic streams since the port numbers would not differ for any specific application. Additionally, pre-classified traffic can be misleading since traffic patterns can show dramatic changes over time. Their traffic classification results shows bulk data, p2p, database, interactive, www, mail, games and multimedia can be classified with a 95% overall accuracy using Fast Correlation-Based Filter (FCBF) methods. However, we have not been able to see any application for VPN or tunneled traffic scenarios for their algorithm.

2) Clustering Algorithms

Clustering techniques has been widely used in research for application classification. One of the earliest papers in this area is proposed by McGregor et al. [7], in which application types such as HTTP, FTP, SMTP, IMAP, NTP and DNS are studied. Instead of classifying each and every traffic type the authors rather used a scheme to identify similar traffic types such as bulk transfer, small transaction and multiple transactions as classification. They used Expectation Maximization [8] algorithm as a clustering technique. However, since their scheme did not try to identify critical traffic types such as VoIP, FTP or WWW it would not be logical to employ this method to real system to accelerate and/or block individual application types such as VoIP.

A paper by Zander and Armitage [9] used the AutoClass [10] algorithm for clustering traffic types such as game traffic, Napster, AOL, HTTP, SMTP, Telnet and FTP traffic. Their classification scheme was application specific however; given a number of application types the scheme they propose found a more number of clusters than application types to be classified. Therefore, there exists a problem of mapping application to a specific cluster which is a critical problem for the efficiency and accuracy of the classifier.

Another proposed approach using a clustering method was proposed by Bernaille et al [11], where the K-means algorithm was employed. The K-means algorithm is a clustering algorithm which clusters the given data using a distance vector scheme. Their classifier was TCP based and therefore other transport protocols such as UDP could not be employed. However, in more general scenario where tunneled traffic is also in use transport protocol information is completely invisible. One other drawback of their proposal was the assumption that their scheme can always get the first few packets of the flow in question. However, this would create issues since in probabilistic methods a packet may be misclassified and initial transaction can be missed. A good classifier should catch the applications at any time. They studied FTP, HTTP, Kazaa, NTP, POP3, SMTP, SSH, HTTPS, POP3S.

One recent paper by Erman et al [12] also used K-means algorithm for clustering and they studied web, p2p and ftp traffic. In their study they assumed that training flows can be identifiable from their payload and header information, again not an option for tunneled traffic.

Researchers up to date studied many application types available in today’s networks. Some chose to classify every application by itself some used a similarity approach and classified in a larger classes. From the application type prospective we can state that file transfer, web, e-mail, p2p and some game traffic type of traffic have been studied. However only a few researchers dealt with encrypted data and some work consisted of application level encryption protocols. Some authors assume that start of each flow can be extracted and some assume that payload and header information can be still used for training phase of the algorithms. However, none of the studies deal with tunneled application types and their identification. Also, work done up to date concentrated on new algorithms and methods for traffic identification. None of the studies tested on a real network environment where any significant improvement has been made on a specific application type such as VoIP. In the most general tunnel case no header or payload information is available for the classifier even in the training case phase.

C. Encrypted Traffic Identification

Regarding encrypted traffic classification, most of the literature focuses on either SSH or SSL traffic rather than IPSec which is the focus of this paper. Due to its challenges this type of traffic identification is in its early stages. For example, Charles et al [13] propose an empirical method based on hidden Markov models, in the tunneled scenarios application types can be identified with up to 20% accuracy. They used simulated tunnels over HTTPS, SSH and AIM.

A paper by Bernaille and Teixeira [14] created a model to detect SSL traffic. They did not include tunneled versions and assume the ability to analyze the first few bytes of SSL traffic before encryption which is not possible for traffic put through a network layer tunnel.

Alshammari and Zincir-Heywood [15] used two supervised machine learning algorithms AdaBoost [16] and RIPPER [18] to detect SSH traffic and classify applications inside SSH. They showed that it is feasible that SSH type traffic can be detected. However they only studied small number applications and dealt with only application layer encryption.

The existing work on encrypted and tunneled identification shows that although quite a few researchers are studying traffic types and encryption there appears to be little work related to classification of network layer tunneled protocols and their application types. This opens a fertile, if rather difficult area of new research.

D. VoIP Traffic Classification in IPSec Tunnels

VoIP traffic is delay and jitter sensitive and QoS is reduced if either get too high. Proposed ML techniques need large computation efforts therefore it is usually not good practice to employ ML for VoIP classification in memory limited devices, or in large proxy devices which are often heavily loaded already.

One of the issues that this paper tries to address is, finding a simple enough algorithm for VoIP identification and prioritization in IPSec tunnels which can be deployed easily to a small network device or proxy server. Research to date mainly concentrates on the identification of traffic types rather than their applicability to a real working system and its benefit such as improvements of QoS of VoIP traffic.

III. ALGORITHM PROPOSAL

The main aim of this paper is to demonstrate that the identification of VoIP will improve VoIP QoS and may slow down other protocols, even to the stage where other protocols become unusable. This could allow a proxy to ensure that network layer tunnels carry VoIP but not other traffic. Unless this is demonstrated to be true then the creation of VoIP detection algorithms for network layer tunnels is a pointless exercise.

An excellent test bed can be made from a simple proxy server which passes VoIP and other traffic from one host to another. This server should first pass just VoIP, and then be stressed with cross traffic as would be the case with real network devices. Next the same tests are performed with a VoIP detection algorithm that allows VoIP to be given priority and other traffic delayed in which case significant performance improvements should be seen to VoIP.

For the purpose of this paper a very simple VoIP detection and control algorithm has been formulated. Very little other network traffic has the packet length of VoIP traffic, typically in the range of 60 to 150 bytes. If a data packet is within this range it will be passed through immediately, otherwise it will be delayed to cause non-VoIP traffic to slow down.

IV. TEST BED AND RESULTS

The implementation used the popular Netfilter libraries and Linux firewall system iptables. The system benefits from the libnetfilter API and it is based on the so called “queue” chain of iptables. Figure 2 demonstrates the basic architecture used.

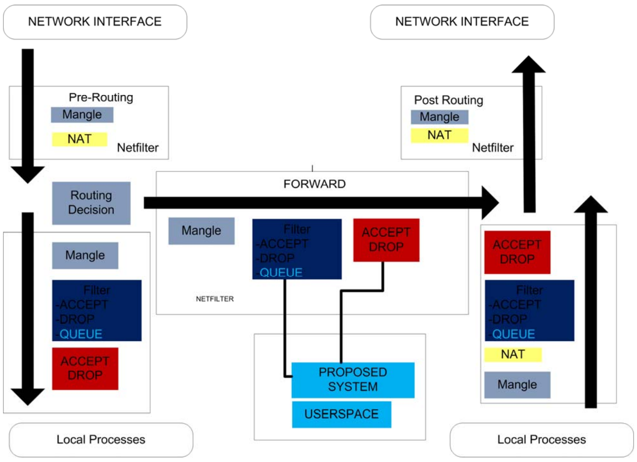

Figure 2. System Level Architecture of Proposed System

It can be seen from the Figure 2 that the key VoIP identifier is placed in user space therefore it can be easily controlled by a network administrator and avoids the need for kernel programming. When a packet comes to the first network interface of a proxy device it runs through the pre-routing phase where it is decided whether the packet is going to be modified or any NAT decision be made. After the routing decision is made it is sent to the FORWARD to be put into a QUEUE which invokes the VoIP identifier. This identifier looks at the packet and its own record of packet history to decide whether or not this packet is most likely a VoIP packet. A VoIP packet is passed straight through the system whereas anything else is queued for 100ms to give priority to VoIP packets. The Proxy of Figure 2 was implemented using GNU/Linux Kernel 2.6 with a 1.86 GHz CPU and 2 Gigabytes of RAM. The proxy device has a standard Gigabit Ethernet controller however host devices have 100 Mbit/s Ethernet controllers.

The test environment examined three distinct scenarios. In the first scenario two host machines are connected through IPSec The second scenario uses the same situation but adds 2.5 megabits of random cross traffic. This is sufficient to seriously stress the proxy system. In the third scenario VoIP/non-VoIP detection is turned on and is used to slow non-VoIP traffic.

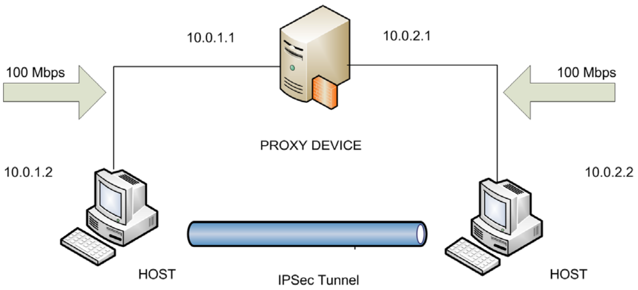

Figure 3. First Test Bed Using a Proxy Device

Table 1 shows an overall view of our results obtained. Our analysis include the packet size of each codec used, average inter arrival times and affect of cross traffic and our proxy system. Note that the heavy cross traffic did not just delay VoIP it actually slowed down the rate at which VoIP traffic is sent out of the proxy. For example 1 second of VoIP traffic on the proxy input would take 1.5 seconds to be transmitted from the proxy output. This situation mimics a network device which is in overload and is dropping packets in order to cope with buffer overflow.

TABLE I. AVERAGE INTER-ARRIVAL TIME OUT OF PROXY

	Inter Arrival Time (seconds)		
Codec	***No Cross Traffic***	***With Cross Traffic***	***Cross Traffic & Proxy***
G711	0.020	0.030	0.020
G723	0.038	0.051	0.040
G729	0.020	0.026	0.021

Above table suggests that cross introduced inter-arrival times increases about 50% which is not an acceptable value in a VoIP call. However, the proxy system successfully decreases the average values to acceptable ranges. Although these values show the significant improvement in the system, these are only average values which cannot analyze the change in jitter conditions which is also an important parameter in VoIP calls. To address that issue we have done a packet analysis for each codec.

In our test environment we used three different codec types G711, G723, G729. As a base test we sent all traffic from 10.0.1.2 to 10.0.2.2 the results are in Figure 4. It was observed that traffic was quite stable and average inter arrival times of each observed packet distributed quite homogenously with low jitter values.

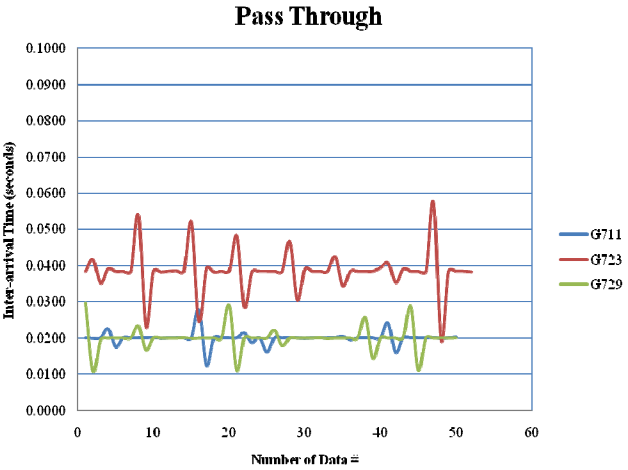

Figure 4. First Test: No external Traffic

Figure 5 shows the distribution of packets when cross traffic is introduced. It is obvious that VoIP traffic is distorted with the cross traffic and inter arrival times vary greatly.

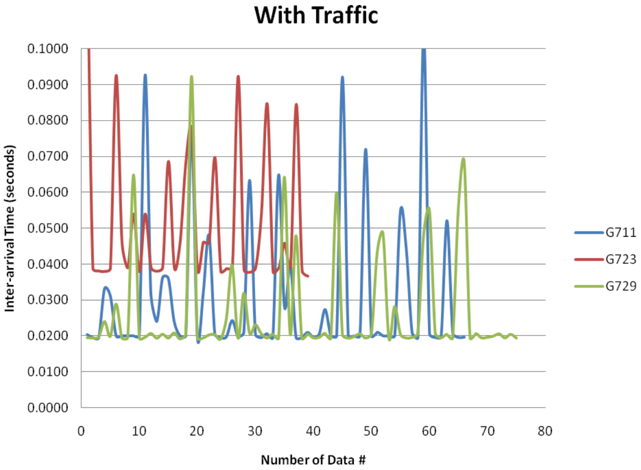

Figure 5. Second Test: With significant cross Traffic

As a final test the proxy system prioritized VoIP traffic when cross traffic is present. In Figure 6 it is obvious that the system decreases the distortions in the packet inter arrival times and prioritize the VoIP traffic.

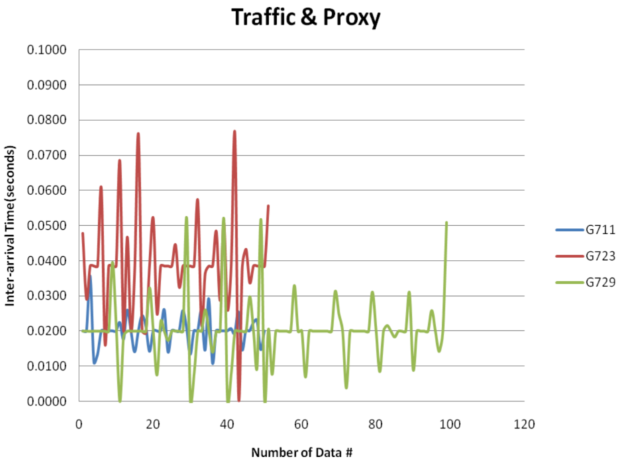

Figure 6. Third Test: with cross traffic and delay for non-VoIP

Figure 7 shows the analysis of G711 codec when there is no cross traffic, when there is cross traffic and when our proxy system works in the presence of cross traffic. It is observed that cross traffic increases the distortion greatly and the proxy system significantly decreases the affect of cross traffic and increases performance.

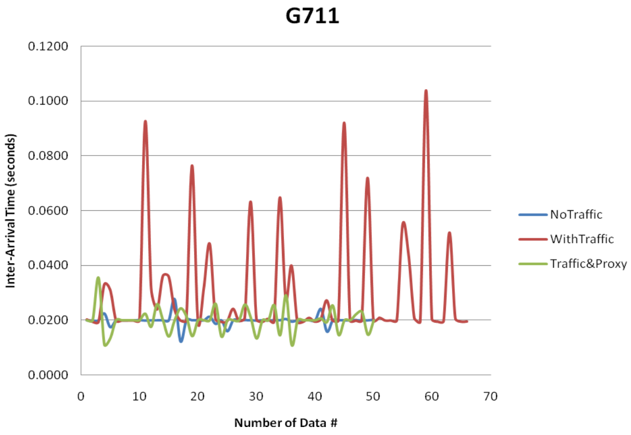

Figure 7. Three Tests for G711 Codec

V. Conclusion

In this paper we have reviewed recent approaches to IP traffic classification with encrypted data flows, including application layer encryption such as SSL and network level encryption such as IPSec. Although ML techniques seem promising and produce good results, implementation of these techniques requires more research in the area of network layer encrypted tunnels. This paper has clearly shown that the ability to identify VoIP in such a tunnel can be of great use for real network devices and so shows that research on traffic identification within network layer tunnels has a practical and important application. Our experimental network shows significant performance increases for network layer tunnelled VoIP with cross traffic when the VoIP component of the tunnel can be identified and given priority

An enterprise such as a school with a proxy that passes network layer tunnels runs the risk of having high traffic to sites or services that would normally be banned. Any VoIP in such a tunnel would have unacceptable QoS. The techniques shown in this paper allow secure VoIP in a tunnel but slow down, and may practically eliminate, non-VoIP traffic.

Traffic identification within an encrypted tunnel also raises a new set of security issues. If the nature of traffic (but not the content) can be identified does this represent a security weakness for the user? Does the user need to respond with counter measures? What counter measures could be used?

Future implementations will include testing novel VoIP identification schemes with a greater variety of cross traffic including real time loads from real enterprises. The current VoIP detection technique uses a simple algorithm for efficiency; however it is likely that a more complex algorithm is needed to cope with peculiar cross traffic and user countermeasures aimed at foiling VoIP/non-VoIP classification. It may also be beneficial to create more complex algorithms to control the delay on non-VoIP traffic to decrease the packet and data loss or make non-VoIP traffic deliberately unusable. These research issues have important practical applications and so are worthy of deeper reflection and consideration.

References

[1] T. T. T Nguyen, G. Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning," IEEE Communications Surveys and Tutorials 4, pp.56-76, 2008.

[2] J. Frank, "Machine learning and intrusion detection: Current and future directions," in Proceedings of the National 17th Computer Security Conference, Washington, D.C., 1994.

[3] M. D. M. Crotti, F. Gringoli, L. Salgarelli, "Traffic Classification through Simple Statistical Fingerprinting," SIGGCOMM Computer Communication Review, vol. 37, pp. 5-16, 2007.

[4] A. W. M. W. Li, "A Machine Learning Approach for Efficient Traffic Classification," in Proceedings of IEEE MASCOTS 2007, pp. 310-317, 2007.

[5] J. R. Quinlan, C4.5: Program for Machine Learning, Morgan Kaufman, 1993.

[6] D. Z. A. W. Moore, "Internet Traffic Classification Using Bayesian Analysis Techniques," SIGMETRICS'05, June 6-10, pp. 50-60, 2005.

[7] M. H. A. McGregor, P. Lorier, J. Brunskill "Flow Clustering Using Expectation Maximation," in Passive and Active Measurement Workshop(PAM2004) Anitbes Juan-les-Pins, France, 2004.

[8] N. L. A. Dempster, D.Rubin, "Maximum likelihood from incomplete data via the EM algorithm," Journal of Royal Statistical Society, vol. 39, pp. 1-38, 1977.

[9] T. N. S. Zander, G. Armitage, "Automated traffic classification and application identification using machine learning," in IEEE 30th Conference on Local Computer Networks (LCN 2005), Sydney, Australia, pp. 250-257, 2005.

[10] J. S. P. Cheeseman, "Bayesian classification (AutoClass): Theory and Results," Advances in Knowledge Discovery and Data Mining, pp.153-180, 1996.

[11] R. T. L. Bernaille, I. Akodkenou, A. Soule, K. Salamatian, "Traffic Classification on the fly," ACM Special Interest Group on Data Communication (SIGCOMM) Computer Communication Review, vol. 36, pp.23-26, 2006.

[12] A. M. J. Erman, M. Arlitt, C. Williamson, "Identifying and discriminating between web and peer-to-peer traffic in network core," in WWW'07:Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, pp. 883-892, 2007.

[13] V. W. Charles, M. Fabian, and M. M. Gerald, "On Inferring Application Protocol Behaviors in Encrypted Network Traffic," J. Mach. Learn. Res., vol. 7, pp. 2745-2769, 2006.

[14] l. Bernaille and R.Teixeira. “Early recognition of encrypted applications,” In Passive and Active Measurement Conference 2007 (PAM’07), pp. 165-175, April 2007.

[15] R. Alshammari and A. N. Zincir-Heywood, "A flow based approach for SSH traffic detection," in Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on, pp. 296-301, 2007.

[16] C. W. W., "Fast effective rule induction," in Proceedings of the 12th International Conference on Learning, pp. 115-123, 1995.

[17] S. R. E. Freund Y., "A Short Introduction to Boosting," Journal of Japanese Society for Artificial Intelligence, vol. 14, pp. 771-780, 1999.