An Investigation on the Identification of VoIP traffic: Case Study on Gtalk and Skype

Riyad Alshammari and A. Nur Zincir-Heywood
Dalhousie University, Faculty of Computer Science
Halifax NS B3H 1W5, Canada
(riyad,zincir)@cs.dal.ca

***Abstract*—The classification of encrypted traffic on the fly from network traces represents a particularly challenging application domain. Recent advances in machine learning provide the opportunity to decompose the original problem into a subset of classifiers with non-overlapping behaviors, in effect providing further insight into the problem domain. Thus, the objective of this work is to classify VoIP encrypted traffic, where Gtalk and Skype applications are taken as good representatives. To this end, three different machine learning based approaches, namely, C4.5, AdaBoost and Genetic Programming (GP), are evaluated under data sets common and independent from the training condition. In this case, flow based features are employed without using the IP addresses, source/destination ports and payload information. Results indicate that C4.5 based machine learning approach has the best performance.**

I. Introduction

The increasingly popular Peer-to-Peer (P2P) Voice over Internet Protocol (VoIP) applications have gain huge success in the last few years and are becoming a major communication service for enterprises and individuals since the cost of VoIP calls is much cheaper than the traditional public Switched Telephone Networks (PSTNs), the voice and video quality is getting better, the communication is free of charge if placed directly from VoIP end user to another one and the dynamic approach to circumvent restrictive network environments such as firewalls and Network Address Translation (NAT) boxes is possible. To date, there are many VoIP products that are able to provide high call quality such as Skype [1], Gtalk [2], Microsoft Messenger (MSN) [3], and Yahoo! Messenger (YMSG) [4].

Thus, an efficient classification of such VoIP traffic represents a fundamental issue for network management tasks such as managing bandwidth budget and ensuring quality of service objectives. Naturally, the process of traffic classification has several unique challenges including: non-standard utilization of ports, embedding of services within encrypted channels, dynamic port-to-application relationships, and the real-time nature of the domain. Usually, Network administrators can depend on well known Transmission Control Protocol (TCP) and/or User Datagram Protocol (UDP) port numbers assigned by the Internet Assigned Numbers Authority (IANA) [5] to classify traffic. However, this approach more and more unreliable, since applications use nonstandard ports to by-pass firewalls or circumvent operating systems restrictions. Moreover, there is no control to force an application to use reserved ports to send or receive traffic. Another extremely accurate approach to classify network traffic is to inspect the payload of every packet. However, encrypted applications such as Gtalk, and Skype imply that the payload is opaque. Thus, other techniques are required to increase the efficiency of VoIP traffic classification.

One possibility is to identify specific features of the VoIP traffic and use these to guide the traffic classification. Recent research in this area focuses on the identification of efficient and effective classifiers. Different research groups have employed expert systems or various machine learning techniques such as Hidden Markov models, Naïve Bayesian models, AdaBoost, or Maximum Entropy methods to this problem [6], [7], [8], [9], [10]. Moreover, the limitations of port and payload based analysis have motivated the use of transport/flow layer statistics for traffic classification [11], [12], [13], [14]. These techniques rely on the observation that different applications have distinct behavior patterns on the network. However, P2P VoIP applications such as Skype and Gtalk adopt a new technique to send and receive traffic using Web port (port 80 or port 443) to by-pass firewall restrictions and traverse NAT boxes.

Skype is a proprietary P2P VoIP application. On the other hand, Gtalk is an instance messenger developed by Google that allows its users to place voice calls, send text messages, check emails and transfer files. Gtalk provides very similar services as of MSN, YMSG and Skype since it has abilities for voice call, instant messaging and buddy lists. In practice, it has resemblance with Skype application since Gtalk encrypts its traffic; however the fundamental protocols and techniques employed are relatively distinctive. Thus, the goal of this work is to develop a model that distinguishes Gtalk/Skype traffic from non-Gtalk/non-Skype traffic without using IP addresses, port numbers or payload information using features based on flow information. In order to identify Gtalk/Skype traffic, three different machine learning algorithms, namely, AdaBoost, C4.5 and GP, are employed. We believe that such an approach can be more robust against evasion attacks if it is successful.

The rest of this paper is organized as follows. Related work is discussed in Section II and an overview of Gtalk and Skype are given in Section III. Section IV presents the machine learning algorithms employed whereas Section V details the data sets and features. The experimental results are presented

in Section VI. Finally, conclusions are drawn and future work is discussed in Section VII.

II. RELATED WORK

To the best of our knowledge, the focus on the literature for detecting VoIP traffic is on Skype traffic. Skype is one of the most commonly used VoIP applications (Skype has 246 million users and around 10 million users are logged in online at any given time [15]). Skype analysis has become popular in the last few years, in part due to the combination of the encrypted operation and dynamic nature of the port assignment making traditional methods of traffic identification redundant. Baset et al. present an analysis of the Skype behavior such as login, NAT and firewall avoidance, and call setting up under three different network conditions [16]. Suh et al. concentrate on the classification of relayed traffic and monitored Skype traffic as an application using relay nodes [17]. Relay node is part of the decentralized Skype network that can ease the routing of Skype traffic to bypass NATs and firewalls. They used several metrics based on features such as inter-arrival time, bytes size ratio and maximum cross correlation between two relayed bursts of packets to detect Skype relay traffic. Their results (a 96% true positive and 4% false positive) show the technique is reliable in recognizing relayed Skype sessions but it might not be appropriate to classify all Skype VoIP traffic. Bonfiglio et al. introduced two approaches to classify Skype traffic [18]. The first approach is to classify Skype client traffic based on Pearson's Chi-Square test using information revealed from the message content randomness (e.g. the FIN and ID fields).Their second approach is to classify Skype VoIP traffic based on Naïve Classifier using packet arrival rate and packet length. They obtained the best results when the first and second approaches were combined. They achieved approximately 1% false positive rate and between 2% to 29% false negative rate depending on the data sets they employed.

On the other hand, we focus on encrypted tunnel identification without using the IP addresses, port numbers and payload data. We have also compared five based classifiers using flow feature set to classify SSH/Skype traffic [19]. Results show that the C4.5 based approach based approach outperforms other algorithms on the data sets employed. Furthermore, recently, we focus on the robustness/generalization of the machine learning based approach to classify Skype encrypted traffic. What we mean by robustness here is that the signatures to classify/identify the encrypted traffic are generated on network traffic from one network but evaluated (tested) on network traffic, which are from completely different networks. We compared Symbiotic Bid-based Genetic Programming (SBB-GP) against C4.5 and AdaBoost, SBB-GP outperforms C4.5 and AdaBoost on identifying Skype encrypted traffic when we evaluated the three machine learning algorithms on different network traces from different institutions using flow based features [20]. Moreover, we have compared (SBB-GP) based classifier against C4.5 [21] on SSH traffic classification using feature based on packet header approach. In that work, results show that GP based classifier was quite competitive with the C4.5 based classifier.

III. SUMMARY OF GTALK AND SKYPE APPLICATIONS

Skype [1] is a very popular P2P VoIP client developed in 2002 by the developers of KaZaa that allows its users to communicate through voice calls, audio conferencing and text messages. Skype protocols are proprietary and an extensive use of cryptography is implemented by the Skype creators. Moreover, Skype employs a number of methods to circumvent NAT and firewall restrictions [16], which increase the difficulty of identifying it. Skype is based on P2P architecture except users authentication, which is performed based on a central architecture. Skype uses the TCP or the UDP protocols at the transport layer to provide its services. For network communication, Skype mostly prefers the UDP protocol. A more detailed description of Skype protocol can be found in [16].

On the other hand, Gtalk application also provides many services to end users, these are: i) voice communication, ii) video communication, iii) file transfer, and iv) chat services. The communication between users is established using a traditional end-to-end IP paradigm, but Gtalk routes call through a relay node to ease the traversal of symmetric NATs and firewalls. Though Gtalk may relay on TCP and UDP at the transport layer, communication data are favorably carried over UDP. Users authentication is performed by a client-server architecture using public key mechanisms. After a user (client) is authenticated, all further communication is carried out with the nearest Google server relay node. This way not only the quality of service can be guaranteed by Google but also both the scaling issues and the control issues can be solved by Google in a more seamless way. The main difference between Gtalk and other VoIP clients is that Gtalk can benefit from the vast amount of Google servers, which is being used as relay nodes (super nodes) to ensure the quality of service. On the other hand, other P2P architectures like Skype can choose any suitable machine on the P2P network as a relay node [16].

IV. CLASSIFIER METHODOLOGIES

In this work, we are interested in the application of supervised machine learning (ML) based techniques to network traffic classification, specifically classification of VoIP traffic. The reason we took a ML based approach is the need for automating the process of identifying such traffic but in terms of automatically creating the signatures (rules) that are necessary to classify VoIP as well as automating the process of selecting the most appropriate attributes for those signatures. The ML techniques require a number of steps. First, a matrix of instances vs. features are needed to describe the data set. A vector of features describes each instance or record in a given trace/traffic file. The features are used as values to quantify different characteristic of the instance (network traffic) such as packet size or inter-arrival time. Second, a label is provided for each instance, which is the class description (network application type). Finally, ML needs to be trained using a data set (called training) and gives an output, which consists of

the rules or the model it generates. This output can then be verified on a test data set (unseen instances). A more detailed explanation of ML and traffic classification can be found in [22].

Given the general success of C4.5 and AdaBoost in previous studies [19], [23], [24], [25], [26], [14], [7], [10], we employ both models during this study in order to establish a performance baseline. A more detailed explanation of C4.5 and AdaBoost algorithms can be found in [27] whereas a more detailed explanation of GP can be found in [28]. The following will summarize the C4.5, AdaBoost and GP algorithms.

A. C4.5

C4.5 is a decision tree based classification algorithm. A decision tree is a hierarchical data structure for implementing a divide-and-conquer strategy of attribute based model building. It is an efficient non-parametric method applicable both to classification and regression. Non-parametric models divide the input space into local regions defined by a distance metric. In a decision tree, the local region is identified in a sequence of recursive splits in smaller number of steps. A decision tree is composed of internal decision nodes and terminal leaves. Each node *m* implements a test function *fm(x)* with discrete outcomes labeling the branches. This process starts at the root and is repeated until a leaf node is encountered. The value of a leaf constitutes the output. In the case of a decision tree for classification, the goodness of a split is quantified by an impurity measure, typically entropy based. Naturally, if the split is not 'pure', then the instances should be split to decrease impurity, and there are multiple possible attributes on which a split can be performed. Such a scheme is locally optimal, hence has no guarantee on finding the smallest decision tree.

B. AdaBoost

AdaBoost, Adaptive Boosting, is a meta-learning algorithm, which means that a strong classifier is built from a linear combination of weak (simple) classifiers. It incrementally constructs a complex classifier by overlapping the performance of possibly hundreds of simple classifiers using a voting scheme. These simple classifiers are called decision stumps. They examine the feature set and return a decision tree with two leaves. The leaves of the tree are used for binary classification and the root node evaluates the value of only one feature. Thus, each decision stump will return either +1 if the object is in class, or -1 if it is out class. AdaBoost is simple to implement and known to work well on very large sets of features by selecting the features required for good classification.

C. Overview of SBB-GP

In this work, we have applied the Symbiotic Bid Based Genetic Programming (SBB-GP) approach to our problem domain. The SBB framework makes extensive use of co-evolution [28], with a total of three populations involved: a population of points, a population of learners, and a population of teams. Individuals comprising a team are specified by the team population, thus establishing a symbiotic relationship with the learner population. Only the subset of individuals indexed by an individual in the team population compete to bid against each other on training exemplars. The use of a symbiotic relation between teams and learners makes the credit assignment process more transparent than in the case of a population wide competition between bids. Thus, variation operators may now be defined for independently investigating team composition (team population) and bidding strategy (learner population). The third population provides the mechanism for scaling evolution to large data sets. In particular the interaction between team and point population is formulated in terms of a competitive coevolutionary relation [29]. As such, the point population indexes a subset of the training data set under an active learning model (i.e. the subset indexed varies as classifier performance improves). Biases are enforced to ensure equal sampling of each class, irrespective of their original exemplar class distribution [30], whereas the concept of Pareto competitive coevolution is used to retain points of most relevance to the competitive coevolution of teams.

V. Evaluation Methodology

As discussed earlier, in this work, the preferred models of classifications from Section IV (C4.5, AdaBoost and SBB-GP) will be evaluated using the flow based feature sets for identifying Gtalk/Skype encrypted traffic from a given network traffic trace. The following describes the testbed setup, traffic generation and features/attributes employed to represent the traffic in this work.

A. Testbed Setup and Traffic Generation

In order to train our ML based classifiers, we needed a controlled environment, where the ground truth is known. Thus, we generated VoIP traffic using different applications on a testbed that we set up. This testbed involved several PCs connected through the Internet and several network scenarios were emulated using Gtalk and other (e.g. Primus, Yahoo messenger) popular VoIP applications. To this end, we observed how Gtalk/Skype reacts to different network restrictions. Moreover, the effects (if any) of different types of access technologies (i.e. WiFi and Ethernet) were also investigated, as well as their combination. Overall, we have conducted over 100 experiments equivalent to more than 25 hours of VoIP traffic. In these experiments, we generated and captured more than 6 GB of traffic at both ends, where approximately 34 million packets were transmitted.

For this work, a Gtalk client was installed on each of the three windows XP machines. The first machine was a Pentium 4 2.4 GHz Core 2 Duo with 2 GB RAM, the second machine was a Pentium 4 2 MHz Core 2 Duo with 2 GB RAM, and the third machine was a MacBook 2 GHz Intel Core 2 Duo with 2 GB RAM. Two machines had a 10/100 Mv/s Ethernet and the third machine had a wireless 10/100 Mv/s card. Furthermore, one was connected to 1 GB/s network while the others were connected to a 10/100 Mb/s network. All three machines had Windows XP Service Packet 2 and all experiments were done

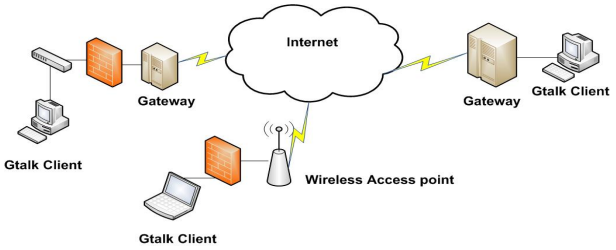

Fig. 1: Network Setup with restrictions.

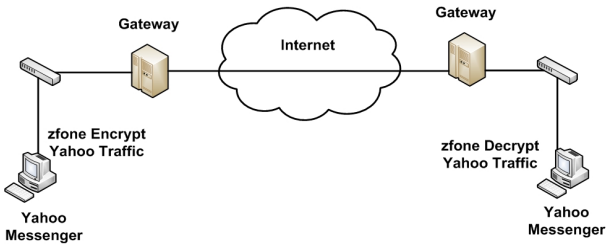

Fig. 2: Network Setup for Zfone Calls.

using the Gtalk client version 1.0.0.104. In all experiments, we have observed the Gtalk behavior from both ends. In all cases, we have performed experiments under several different network scenarios, Figure 1.

These scenarios include: i) Firewall restrictions on one user end and no restriction at the other end; ii) Firewall restrictions at both users ends; iii) No restrictions at both users ends; iv) Use of wireless and wire-line connections; v) Blocking of all UDP connections, and vi) Blocking of all TCP connections. It should be noted here that during these experiments all the Internet communications went through our networks firewall. The firewall was configured to permit access to the aforementioned restrictions such as do not permit anything, or permit limited well known port numbers such as port 22, 53, 80 and 443. Moreover, we have observed the Gtalk client through out the installation period as well as the first time login. Wireshark [31] and NetPeeker [32] were used to monitor and control network traffic. NetPeeker was used to block ports and to allow either both TCP and UDP traffic, or only UDP or TCP traffic in order to analyze the behavior of the Gtalk client. On the other hand, Wireshark was used to capture traffic from both users ends.

The general call set up between the caller and callee for voice calls is as follows: caller transmits a standard audio file to callee. We used an English spoken text (male and female audio files) without noise and a sample rate of 8 kHz, which was encoded with 16 bit per sample and can be downloaded at [33]. The wav-file was played and then the output of Windows media player was used as input for Gtalk, Primus (soft Talk Broadband (softTBB)) and Yahoo messenger (Encrypted with zfone) clients using a microphone. Wireshark was used to capture the traffic from both users' ends. We have made this testbed traffic publicly available to the research community, too [34].

On the other hand, we have also generated Yahoo messenger traffic (encrypted with Zfone) and Primus VoIP traffic as well as online banking traffic in order to distinguish Gtalk and Skype traffic from these similar applications. In these experiments, we generated and captured more than 6 GB of traffic at both ends, where approximately 34 million packets were transmitted.

Furthermore, Zfone traffic is another encrypted VoIP traffic we generated. Zfone [35] is a software that secures VoIP calls over the Internet. Zfone works by intercepting all the unencrypted VoIP channel and securely protect the VoIP channel by encrypting all the VoIP packets. Zfone is the user interface of the ZRTP protocol [36]. ZRTP uses a Diffie-Hellman to exchange key over RTP (Real-time Transport Protocol) packet stream (creates secure RTP sessions). It encrypts the payload of a packet using standard cryptographic algorithms such as Advanced Encryption Standard (AES) or Rivest Shamir Adleman (RSA) algorithms. We used Zfone to secure all Yahoo Messenger audio calls. Zfone detects Yahoo packets and encrypts them as they are sent by the caller machine and detects the encrypted packets received by the callee machine and decrypts them, Figure 2.

On the other hand, we also generated non-encrypted VoIP traffic using Primus Session Initiation Protocol (SIP) client [37]. Primus Enterprise VoIP network deploys the SIP [38] to set up, validate and complete calls over the Internet. The main components of the Primus Enterprise VoIP network are: i) IP phone: a terminal (softTBB software) with native VoIP support and direct connection to the Internet; ii) Primus Voice Gateway with ability to convert network signals from/to the telephony interfaces and the VoIP protocols; and iii) Primus SIP server, which is responsible in providing the management and administrative functions with the essential support to route calls across the network. We used the Primus softTBB to make calls to Public Switched Telephone Network (PSTN) for voice services (hard line phone) and Bell [39] Mobil Cell phone. The softTBB client runs on a PC or a laptop and connects to the Primus SIP Network over the Internet. Depending on what we call, i.e. a mobile phone or a PSTN phone, Primus SIP network routes the calls to the final destination differently. For a call to a PSTN phone, the calls are routed to the nearest Primus Voice gateway, which is responsible for making the communication (converting the calls) between the VoIP network and PSTN network. On the other hand, for a call to a cell phone, which is subscribed to the Bell General Packet Radio Service (GPRS) and Universal Mobile Telecommunication System (UMTS) network, the route is more complex. According to GPRS/UMTS specification [40], the mains components of the GPRS/UMTS network are base stations and gateways connected to the Internet. In this case, firstly, the cell phone registers with the base station. Then, the base station is connected to the Serving Gateway Support Node (SGSN), which is connected to the Gateway GPRS Support Node (GGSN) inside the Bell GPRA/UMTS network. Finally,

the GGSN is the first node that is responsible for processing IP packets from the Internet to the mobile network and vice versa. To establish the call between a Primus softTBB and a Bell mobile phone, the call is routed through the Primus SIP network through the Internet to the Bell GGSN gateway. In all cases, we were able to listen to the call at the PSTN phone and the mobile phone. All communications are done without encryption and the traffic is captured using Wireshark only at the machine, where the softTBB is running, since we do not have permission to capture traffic with Primus or Bell companies. In this case, we deliberately choose not to encrypt the traffic so that we have different mixtures of VoIP traffic in our traces, i.e. both encrypted (Gtalk, Skype, and Yahoo with Zfone) and non-encrypted (Primus, Yahoo, IM). Furthermore, we have captured an encrypted online banking traffic, which is also included in these traffic traces.

Last but not the least, we have also employed network traces captured on the campus network of our university. To this end, university traces employed in this work contain DNS, FTP, SSH, MAIL, HTTP, HTTPS and MSN traffic. Thus, we have traffic traces of 11 applications that have similar behavior to Gtalk. In short, we believe that the traffic traces we employed in this work are representative of traces that can be encountered in real life. It should be noted here that University traffic traces were captured on the Dalhousie University Campus network by the University Computing and Information Services Centre (UCIS) in January 2007. Dalhousie is one of the biggest universities in the Atlantic region of Canada. There are more than 15000 students and 3300 faculty and staff. The UCIS is responsible for all the networking on the campus, which includes more than 250 servers and 5000 computers. Moreover, the wireless network is enabled on the campus, where thousands of users (students and staff) are connected daily. Dalhousie network is connected to the Internet via a full-duplex T1 fiber link. Full-duplex traffic on this connection was captured for 8 hours. Given the privacy related issues, data is filtered to scramble the IP addresses and each packet is further truncated to the end of the IP header so that the payload is excluded. Moreover, the checksums are set to zero since they could conceivably leak information from short packets. However, any information regarding size of the packet is left intact. The University traces are labeled using a commercial classification tool (PacketShaper), which is a deep packet analyzer [41], by the university's network team, UCIS (i.e. not by us). PacketShaper uses Layer 7 filters (L7) to classify applications [42].

Finally, establishing the ground truth for the traces (Gtalk, Primus etc.) that we generated on our testbed was not a problem, since we knew exactly which applications were running in every experiment. Brief statistics on the traffic data collected are given in Table I. In this work, for Gtalk/Skype traffic identification, we have used a sampled subset of Gtalk traces and mix them with University traces as the training data set. Naturally, the rest of the University traces, Zfone traces, Primus traces, online banking traces and the rest of the Gtalk traces are used as the testing data set. In total, Test traces consist of 44,588,269 flows.

TABLE I: An overview of network traces employed

	University	**Gtalk**	**Primus**	**Zfone**
Total Packets	337,041,778	34,292,124	1,920,991	1,092,093
MBytes	213,562	6,492	384	146
% of TCP packets	86.51%	18.70%	0.06%	1.32%
% of TCP bytes	91.03%	17.0%	0.18%	0.99%
% of UDP packets	13.33%	81.30%	99.94%	98.68%
% of UDP bytes	8.95%	83%	99.82%	99.01%
% of Other packets	0.16%	0.0%	0.0%	0.0%
% of Other bytes	0.02%	0.0%	0.0%	0.0%
# non Gtalk Flows	35,565,561	0	9,591	29,961
# Gtalk Flows	0	301,735	0	0
# Skype Flows	8,664,137	0	0	0

TABLE II: Flow based feature set employed

	Feature Name	**Abbreviation**
1	Protocol	proto
2	Duration of the flow	Duration
3	# Packets in forward direction	fpackets
4	# Bytes in forward direction	fbytes
5	# Packets in backward direction	bpackts
6	# Bytes in backward direction	bbytes
7	Min forward inter-arrival time	minfiat
8	Mean forward inter-arrival time	meanfiat
9	Max forward inter-arrival time	maxfiat
10	Std deviation of forward inter-arrival times	stdfiat
11	Min backward inter-arrival time	minbiat
12	Mean backward inter-arrival time	meanbiat
13	Max backward inter-arrival time	maxbiat
14	Std deviation of backward inter-arrival times	stdbiat
15	Min forward packet length	minfpkt
16	Mean forward packet length	meanfpkt
17	Max forward packet length	maxfpkt
18	Std deviation of forward packet length	stdfpkt
19	Min backward packet length	minbpkt
20	Mean backward packet length	meanbpkt
21	Max backward packet length	maxbpkt
22	Std deviation of backward packet length	stdbpkt

B. Feature Selection -Flow based Features

The flow based feature set, a feature is a descriptive statistic that can be calculated from one or more packets for each flow. To this end, NetMate [43] is employed to generate flows and compute 22 feature values, Table II. Flows are bidirectional and the first packet seen by the tool determines the forward direction. In this work, we consider only UDP and TCP flows. Moreover, UDP flows are terminated by a flow timeout, whereas TCP flows are terminated upon proper connection teardown or by a flow timeout, whichever occurs first. The flow timeout value employed in this work is 600 seconds as recommended by the IETF [44].

VI. Empirical Evaluation

In traffic classification, two metrics are typically used in order to quantify the performance of the classifier: Detection Rate (DR) and False Positive Rate (FP). In this case, DR will reflect the number of Gtalk/Skype flows correctly classified and is calculated using $DR = \frac{TP}{TP+FN}$; whereas FP rate will reflect the number of non-Gtalk/non-Skype flows

TABLE III: SBB-GP parameters

Parameter	Description	Value
P_{size}	Point population size.	90
M_{size}	Team population size.	90
t_{max}	Number of generations.	30000
p_d	Probability of learner deletion.	0.1
p_a	Probability of learner addition.	0.2
μ_a	Probability of learner mutation.	0.1
ω	Maximum team size.	30
P_{gap}	Point generation gap.	30
M_{gap}	Team generation gap.	60

incorrectly classified as Gtalk/Skype and is calculated using $FPR = \frac{FP}{FP+TN}$. Naturally, a high DR rate and a low FP rate are the most desirable outcomes. Moreover, False Negative, FN, implies that Gtalk/Skype traffic is classified as non-Gtalk/Skype traffic, and False Positive, FP, implies that non-Gtalk/non-Skype traffic is classified as Gtalk/Skype traffic.

All three candidate classifiers are trained on the training data using fifty runs to generate 50 different models for each run so that the results are statistically valid. Weka [45] is employed with default parameters to run C4.5 and AdaBoost. Fifty runs of the C4.5 algorithm are performed using different confidence factors to generate different models for C4.5 and fifty runs of the AdaBoost algorithm are performed using different weight thresholds to generate different models for AdaBoost. The SBB-GP classifier's default parameters are summarized in Table III. Fifty runs of the SBB-GP algorithm are performed using different population initializations to generate different models.

A. Results

In these set of experiments, the objective is to identify Gtalk and Skype on a flow by flow basis using only the set of features given in Table II. To this end, we first trained each classifier on our training data set using the same feature set. Then, we tested each trained model (C4.5, AdaBoost and SBB-GP) on the test data set, which consists of five test traces, namely, University traces, Gtalk, Primus, Zfone and Italy traces.

Results given in Figures 3 and 4 illustrate that C4.5 based classification approach is much better than other algorithms employed in identifying the Skype flow traffic based on the training data set. Moreover, in the case of C4.5, much lower variance (Table IV) implies that the corresponding solutions generalize to the wider case, implicit in the test results. We use these trained models, on all of the complete traces employed. Furthermore, Table I shows that the percentages of the TCP and UDP traffic are different for each trace. What this demonstrates is that these traces indeed belong to substantially different networks. Therefore, we believe that only well generalized models are able to classify Skype or Gtalk traffic correctly on these networks.

To visualize which machine learning algorithms have the most diverse performance on the test data sets, Figures 5 and 6 show the DR and FPR for all 50 models on the test data sets. On average, C4.5 is much better than other machine

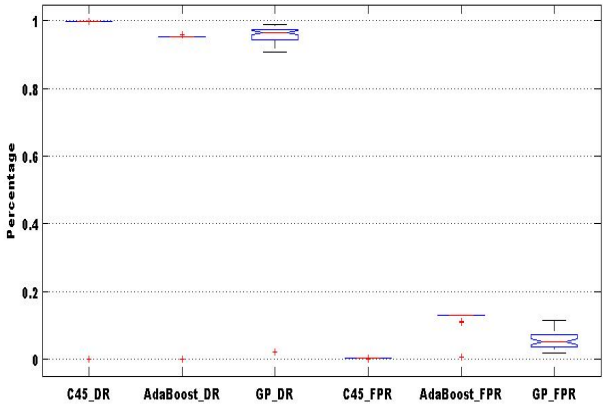

Fig. 3: Results on the Training Data set for Flow based Feature set for Skype detection.

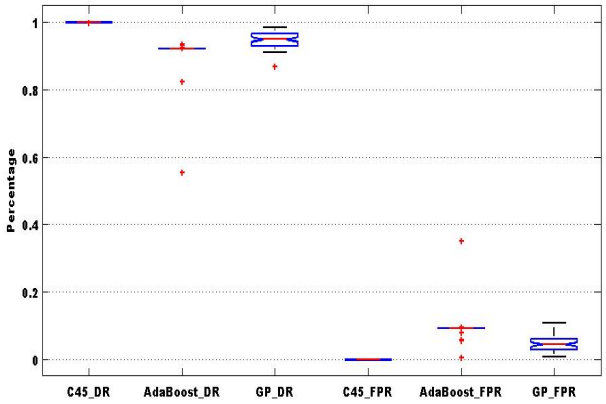

Fig. 4: Results on the Training Data set for Flow based Feature set for Gtalk detection.

learning algorithms on the test data sets in terms of high DR and low FPR. In the case of Skype, C4.5 based classification approach is much better than other machine learning algorithms employed in identifying the Skype traffic. C4.5 based system can correctly classify ≈99% of the instances with less than 1% FPR on combined test traces. For Gtalk, results show that again, the C4.5 classifier performs better than the other classifiers on all of the data sets. C4.5 classifier achieves ≈99% DR and 0.2% FPR on combined test traces. Moreover, the SBB-GP classifier is very competitive with C4.5 under test traces (particularly in terms of Gtalk False Positive rate and Gtalk Detection rate) whereas the AdaBoost based system performs the poorest of the three.

Furthermore, we note that SBB-GP generates fewer individuals (rules) on average to detect Gtalk/Skype flow traffic (on average 10 individuals for both Skype and Gtalk detection), Figures 7 and 8. Moreover, C4.5 found the most complex solution for Skype (on average 350 rules for Skype detection

TABLE IV: Standard Deviation of Results on the training data for Gtalk/Skype based on flow feature set

	C4.5		AdaBoost		GP	
	DR	**FPR**	**DR**	**FPR**	**DR**	**FPR**
Training Sample (subset of university) x 50						
Non-Skype	0.0002	0.0001	0.0046	0.0008	0.026	0.0214
Skype	0.0001	0.0002	0.0008	0.0046	0.0214	0.026
Non-Gtalk	2.6E-05	0.0001	0.096	0.137	0.023	0.024
Gtalk	0.0001	2.6E-05	0.137	0.096	0.024	0.023

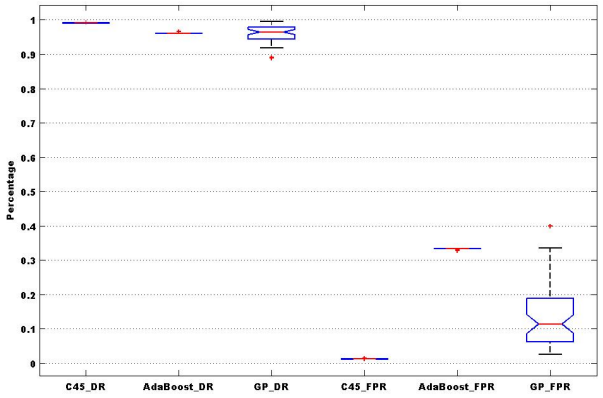

Fig. 5: Results on all Test Data sets for Flow based Feature set for Skype detection.

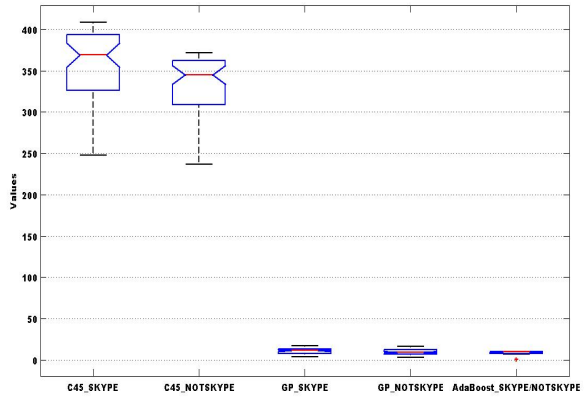

Fig. 7: Number of Rules/Individuals for each classifier for Skype detection.

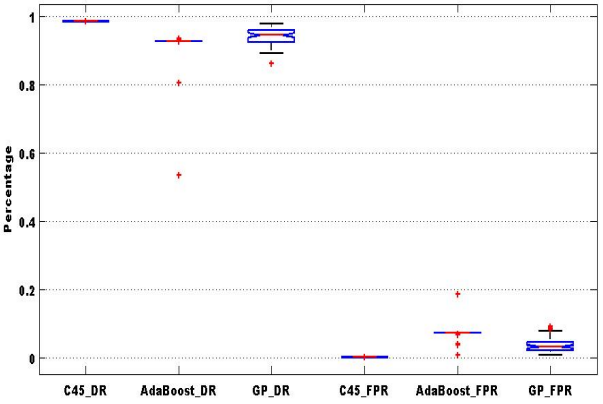

Fig. 6: Results on all Test Data sets for Flow based Feature set for Gtalk detection.

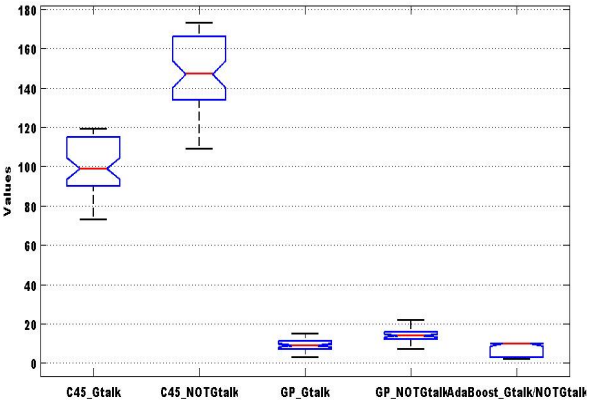

Fig. 8: Number of Rules/Individuals for each classifier for Gtalk detection.

and 100 rules for Gtalk detection).

Figures 9 and 10 list the application flows, which C4.5 misclassifies as Gtalk/Skype flows for the traces employed. Given the fact that both Gtalk and Skype uses many services such as DNS, HTTP, HTTPS, and STUN, and the fact that both of them can run over TCP or UDP, this is to be expected. On the other hand, the flow based classifier can sometimes classify HTTP, HTTPS, Skype and Zfone flows as Gtalk flows since these applications can be used to transfer data between two hosts and some of them are in fact encrypted traffic as well. In the case of Skype detection, C4.5 and SBB-GP flow based classifiers are mostly classifying DNS traffic as Skype since Skype can use UDP protocol to set up communication calls. As a final note, even though there were not many flows representing the Other class in the training data sets, results show that a few of them misclassified during the tests, so we can say that the C4.5 classifier was able to generalize well on that front.

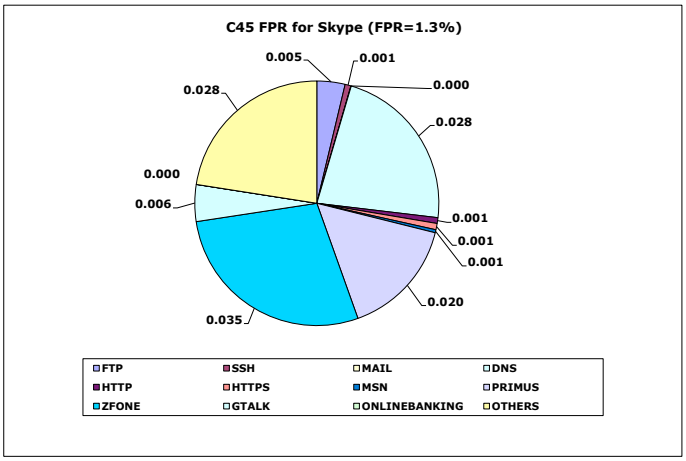

Fig. 9: Applications that are mostly misclassified as Skype by the C4.5 model based on the flow feature set.

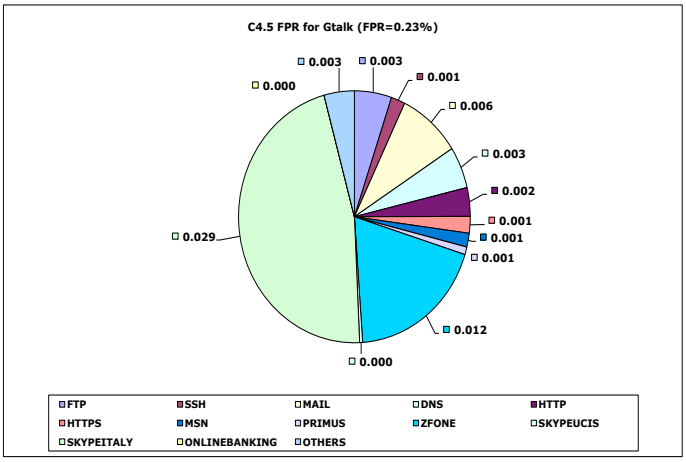

Fig. 10: Applications that are mostly misclassified as Gtalk by the C4.5 model based on the flow feature set.

VII. CONCLUSION AND FUTURE WORK

In this work, we have evaluated three machine learning algorithms, namely AdaBoost, SBB-GP and C4.5, for classifying VoIP traffic in particular Gtalk and Skype traffic from a given traffic file. In this case, the classification based approach is employed with flow attributes.

In our experiments, the C4.5 based classifier can achieve a ≈99% DR and less than ≈1% FPR at its best test performance using the flow based feature set to detect Gtalk and Skype traffic. It should be noted again that in this work, automatically identifying VoIP traffic from a given network trace is performed without using any payload, IP addresses or port numbers. Thus, the automatic rules, i.e. solutions, generated by C4.5 are robust generic signatures as well as being easy to understand.

To the best of our knowledge, there is no publicly available data set that includes Gtalk traffic. Thus, we generated our own Gtalk traffic traces for this research. To make the training and test data sets as realistic as possible, we included other VoIP (Skype, Yahoo, Primus) data as well as other encrypted data (SSH, Skype, Yahoo with Zfone). Moreover, we have included other well-known TCP and UDP based applications in our traffic traces. In summary, we believe that our experimental results show a very promising performance for our classifiers to identify and differentiate VoIP traffic from other encrypted or non-encrypted traffic.

Future work will follow similar lines to perform more tests on different and larger data sets in order to continue to evaluate the robustness of the proposed approach. Moreover, as a next step, we aim to train and evaluate classifiers for other encrypted applications.

ACKNOWLEDGMENT

This work was supported by MITACS grant. Special thanks to Dana Echtner for helping us in setting up the Google Talk experiments. All research was conducted at the Dalhousie University, Faculty of Computer Science NIMS Laboratory, http://www.cs.dal.ca/projectx.

REFERENCES

[1] Skype, http://www.skype.com/useskype/.
[2] "Google talk (gtalk)," last accessed October, 2009, http://www.google.com/talk/.
[3] "Msn messenger," last accessed October, 2009, http://webmessenger.msn.com/.
[4] "Yahoo messenger," last accessed October, 2009, http://messenger.yahoo.com/.
[5] I. A. N. A. (IANA), last accessed October, 2009, http://www.iana.org/assignments/port-number.
[6] C. Wright, F. Monrose, and G. M. Masson, "HMM profiles for network traffic classification," in *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. New York, NY, USA: ACM Press, 2004, pp. 9–15.
[7] P. Haffner, S. Sen, O. Spatscheck, and D. Wang, "ACAS: automated construction of application signatures," in *MineNet '05: Proceeding of the 2005 ACM SIGCOMM workshop on Mining network data*. New York, NY, USA: ACM Press, 2005, pp. 197–202.
[8] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *SIGMETRICS '05: Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*. New York, NY, USA: ACM Press, 2005, pp. 50–60.
[9] A. W. Moore and K. Papagiannaki, "Toward the accurate identification of network applications," in *Passive and Active Network Measurement: Proceedings of the Passive &Active Measurement Workshop*, 2005, pp. 41–54.
[10] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 5, pp. 5–16, 2006.
[11] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM Press, 2005, pp. 229–240.
[12] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian, "Traffic classification on the fly," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 2, pp. 23–26, 2006.
[13] J. Erman, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *MineNet '06: Proceedings of the 2006 SIGCOMM workshop on Mining network data*. New York, NY, USA: ACM Press, 2006, pp. 281–286.
[14] J. Early, C. Brodley, and C. Rosenberg, "Behavioral authentication of server flows," in *Proceedings of the 19th Annual Computer Security Applications Conference*, 2003, pp. 46–55.

[15] "Skype reaches 10 million concurrent users," last accessed May, 2010, http://seekingalpha.com/article/50328-ebay-watch-59-earnings-growth-skype-reaches-10-million-concurrent-users.

[16] S. A. Baset and H. G. Schulzrinne, "An analysis of the skype peer-to-peer internet telephony protocol," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, April 2006, pp. 1–11.

[17] D. K. Suh, D. R. Figueiredo, J. Kurose, and D. Towsley, "Characterizing and detecting relayed traffic: A case study using skype," *in INFOCOM 06: Proceedings of the 25th IEEE International Conference on Computer Communications*, Apr 2006.

[18] D. Bonfiglio, M. Mellia, M. Meo, D. Rossi, and P. Tofanelli, "Revealing skype traffic: when randomness plays with you," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 37–48, 2007.

[19] R. Alshammari and A. N. Zincir-Heywood, "Machine learning based encrypted traffic classification: Identifying ssh and skype," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, July 2009, pp. 1–8.

[20] R. Alshammari and N. Zincir-Heywood, "Unveiling skype encrypted tunnels using GP," in *2010 IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE. WCCI 2010*, July 2010.

[21] R. Alshammari, P. I. Lichodzijewski, M. Heywood, and A. N. Zincir-Heywood, "Classifying ssh encrypted traffic with minimum packet header features using genetic programming," in *GECCO '09: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference*. New York, NY, USA: ACM, 2009, pp. 2539–2546.

[22] T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *Communications Surveys Tutorials, IEEE*, vol. 10, no. 4, pp. 56 –76, fourth 2008.

[23] R. Alshammari and A. Zincir-Heywood, "A preliminary performance comparison of two feature sets for encrypted traffic classification," *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08*, pp. 203–210, 2008.

[24] R. Alshammari and A. N. Zincir-Heywood, "Investigating two different approaches for encrypted traffic classification," in *PST '08: Proceedings of the 2008 Sixth Annual Conference on Privacy, Security and Trust*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 156–166.

[25] R. Alshammari, A. N. Zincir-Heywood, and A. A. Farrag, "Performance comparison of four rule sets: An example for encrypted traffic classification," *Privacy, Security, Trust and the Management of e-Business, World Congress on*, vol. 0, pp. 21–28, 2009.

[26] R. Alshammari and N. Zincir-Heywood, "Generalization of signatures for ssh encrypted traffic identification," in *Computational Intelligence in Cyber Security, 2009. CICS '09. IEEE Symposium on*, 30 2009-April 2 2009, pp. 167–174.

[27] E. Alpaydin, *Introduction to Machine Learning*. MIT Press, 2004.

[28] P. Lichodzijewski and M. I. Heywood, "Managing team-based problem solving with Symbiotic Bid-based Genetic Programming," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2008, pp. 363–370.

[29] E. de Jong, "A monotonic archive for pareto-coevolution," *Evolutionary Computation*, vol. 15, no. 1, pp. 61–93, 2007.

[30] J. Doucette and M. Heywood, "Gp Classification under Imbalanced Data Sets: Active Sub-sampling and AUC Approximation," in *European Conference on Genetic Programming*, ser. Lecture Notes in Computer Science, vol. 4971, 2008, pp. 266–277.

[31] Wireshark, Last accessed Sep, 2008, http://www.wireshark.org/.

[32] "Net peeker," last accessed October, 2009, http://www.net-peeker.com.

[33] "Signalogic, speech codec wav samples," last accessed October, 2009, http://www.signalogic.com/index.pl?page=codec_samples.

[34] "Gtalk data set," last accessed March, 2010, http://www.cs.dal.ca/˜riyad/.

[35] "The zfone project," last accessed October, 2009, http://zfoneproject.com/getstarted.htm.

[36] E. P. Zimmermann, A. Johnston and J. Callas, "Zrtp: Media path key agreement for secure rtp," January 2010, http://tools.ietf.org/html/draft-zimmermann-avt-zrtp-17.

[37] "Primus softphone client," last accessed October, 2009, http://www.primus.ca/en/residential/talkbroadband/talkBroadband-softphone.htm.

[38] J. Rosenberg and H. Schulzrinne, "Sip: Session initiation protocol," June 2002, http://www.ietf.org/rfc/rfc3263.txt.

[39] "Bell canada," last accessed October, 2009, http://www.bell.ca.

[40] "Digital cellular telecommunications system (phase 2+), general packet radio service (gprs), overall description of the gprs radio interface, stage 2 (gsm 03.64, version 7.0.0, release 1999)."

[41] PacketShaper, last accessed March, 2008, http://www.packeteer.com/products/packetshaper/.

[42] l7 filter, last accessed March, 2008, http://l7-filter.sourceforge.net/.

[43] NetMate, http://www.ip-measurement.org/tools/netmate/.

[44] IETF, http://www3.ietf.org/proceedings/97apr/97apr-final/xrtftr70.htm.

[45] "WEKA software," http://www.cs.waikato.ac.nz/ml/weka/.