

Rapid Identification of Skype Traffic Flows

Philip A. Branch
Centre for Advanced Internet
Architectures
Faculty of ICT
Swinburne University of
Technology
Melbourne, Australia
pbranch@swin.edu.au

Amiel Heyde
Centre for Advanced Internet
Architectures
Faculty of ICT
Swinburne University of
Technology
Melbourne, Australia
aheyde@swin.edu.au

Grenville J. Armitage
Centre for Advanced Internet
Architectures
Faculty of ICT
Swinburne University of
Technology
Melbourne, Australia
garmitage@swin.edu.au

ABSTRACT

In this paper we present results of experimental work using machine learning techniques to rapidly identify Skype traffic. We show that Skype traffic can be identified by observing 5 seconds of a Skype traffic flow, with recall and precision better than 98%. We found the most effective features for classification were characteristic packet lengths less than 80 bytes, statistics of packet lengths greater than 80 bytes and inter-packet arrival times. Our classifiers do not rely on observing any particular part of a flow. We also report on the performance of classifiers built using combinations of two of these features and of each feature in isolation.

Categories and Subject Descriptors

C.2.3 [Network Operations]: Network Monitoring, Public Networks

General Terms

Algorithms, Measurement, Experimentation

Keywords

Skype, Traffic classification, Machine learning

1. INTRODUCTION

Skype is a popular, proprietary Voice over IP (VoIP) application. Its operators claim that there are over 400 million users. Skype is peer-to-peer, communication is encrypted and the details of its protocol structure are secret [2][3][4][5].

Our interest in Skype is in developing techniques for rapidly identifying it. We are interested in being able to examine a short sequence of packets that are part of a much larger flow and answering the question “Is this a Skype flow?” Others have investigated using machine learning to identify Skype [4][5][6], but in the context of having access to the full flow and with no

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

NOSSDAV'09, June 3–5, 2009, Williamsburg, Virginia, USA.
Copyright 2009 ACM 978-1-60558-433-1/09/06...\$5.00.

obligation of doing the classification in realtime or near-realtime. In this paper we show how classification can be carried out rapidly (within a few seconds) by observing any part of a flow.

Our interest in this area is motivated primarily by lawful interception obligations [7][8][9][10][11]. Recently governments have been clarifying the ISP obligations with respect to lawful interception. Just as telephone companies are obliged to support interception of telephone communications, so Internet Service Providers (ISPs) are increasingly being obliged to provide information about network usage by particular individuals. Despite the difficulties in doing so [11], the United States government has extended the Communications Assistance for law enforcement agencies Act (CALEA) to be extended to Internet based services, making ISPs subject to the same interception obligations as conventional telecommunications companies [7].

Because it is encrypted and peer-to-peer, Skype presents great challenges to lawful interception. One possibility is for Skype (the organization) to intercept communications. However, there are obvious jurisdictional and administrative issues with this approach.

Ultimately, it is likely that the onus for interception of Skype traffic will fall upon the ISP [8][9]. Lawful interception practice usually consists of a law enforcement agency issuing a warrant instructing the service provider to provide information about and possibly the contents of communication between parties of interest [10]. Given Skype’s peer-to-peer nature and jurisdictional difficulties, the easiest point through which Skype traffic can be intercepted is the user’s ISP.

Fortunately, law enforcement agencies are often only interested in who is talking to whom, rather than in what they are saying [8]. Consequently, it may well be sufficient if the ISP provides to the law enforcement agencies information such as source and destination IP addresses about flows it knows to be Skype.

Identifying traffic flows has, in the past usually been done based on well known port numbers [12]. However, the effectiveness of this method is rapidly declining as applications using unregistered or random ports, particularly peer-to-peer applications, have proliferated. A more recent method of identifying application type associated with a particular flow has been deep packet inspection. A protocol state machine uses information contained within the packet to identify the type of the application. Again, because it uses random ports and its payload is encrypted, neither of these methods works well with Skype [4]. Differentiating applications

by comparing statistical properties such as average packet length and interarrival times provide alternative approaches to identifying the application without the need to rely on port numbers or on inspecting the packet contents [12][19][21][22]. We adopt this approach in the research described in this paper.

Although our interest in Skype is driven by our interest in lawful interception, identification of Skype traffic may be useful in other ways as well. For example, it could be useful in market research, enabling the gathering of detailed statistics as to how much Skype traffic is transmitted across an ISP's network. Another application might be with Telecommunications companies who run both conventional telephony systems and an ISP and who wish to degrade service given to Skype. More positively, there is also the possibility of an ISP providing quality of service guarantees to Skype users. If a subscriber to an ISP requests it (perhaps paying a premium price) the ISP may implement priority queuing mechanisms for giving real-time traffic from that user (such as Skype) priority on its network [20].

In this paper we investigate how Skype traffic can be reliably and rapidly identified from observing the statistical properties of a small sequence of any part of a flow. We show that making use of techniques of machine learning we can identify Skype traffic reliably with only a few seconds of traffic. We adopt the approach described in [21] and [22] where characteristics of a flow are extracted from short sliding windows and used to train a classifier.

The paper is structured as follows. Section 2 provides some additional information about the Skype application. Section 3 is an overview of using machine learning techniques in IP traffic flow identification. Section 4 describes how we carried out the construction and testing of our Skype flow classifiers. Section 5 reports on our results while Section 6 is our conclusion.

2. SKYPE

Despite the secrecy surrounding Skype's internals, investigations have enabled a number of its characteristics to be identified [1][4][5]. It uses a number of codecs, the most popular currently being the Sinusoidal Voice Over Packet Coder (SVOPC) [13]. As of version 3.2 this was the default Skype codec. For this reason, this work focuses on the statistical properties of Skype traffic when using the SVOPC codec.

SVOPC is a variable bit rate codec that varies its rate according to which party is talking and adapts to the available bandwidth [13].

Skype signaling, along with the rest of the application is quite obscure. Nevertheless, the process of setting up a call appears to involve communication from the Skype Client to a Supernode, (another Skype user with adequate free bandwidth and a public IP address). The Supernode then acts as a relay for transmission of authentication information between the Skype Client and the login server.

The Skype Client will attempt to use UDP as transport for voice data, using a variant of the STUN protocol where necessary, to operate behind a firewall with Network Address Translation (NAT). If attempts to send and receive voice directly via UDP are unsuccessful, both clients participating in the call will connect to a Skype Supernode. The Supernode is then used as a relay to enable conversations between the two end-points of the call. Should a firewall block UDP traffic then Skype will use TCP as a

transport protocol. Figure 1 illustrates the main nodes involved in Skype. In our proposals we would intend locating the classifier at one or both of the ISPs.

Using machine learning to identify Skype traffic has attracted some interest recently [4][6]. In [4] the authors were able to obtain close to 100% accuracy in identifying Skype traffic, using

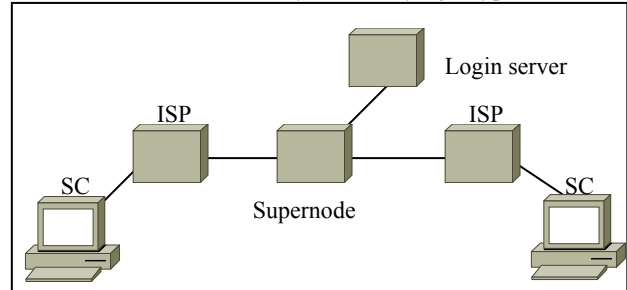


Figure 1. Skype environment

Naïve Bayesian classifiers. However, their method relied on having the full flow for examination and being able to select blocks of data from specified parts of the flow as input to their classifier. It is not intended to be a realtime classification scheme. In [6] the authors were interested in identifying traffic that indicated a particular node was unwittingly being used as a Skype relay node. Again, they achieved very high levels of accuracy but their purpose was to identify relayed Skype traffic, not all Skype traffic flows. As far as we are aware, our work is the first to look at using machine learning techniques to identify Skype in near realtime.

3. USING MACHINE LEARNING FOR TRAFFIC IDENTIFICATION

Machine learning has been shown to be quite effective at IP traffic classification tasks[12]. Using machine learning to identify particular classes of traffic involves the following steps [14]. First, characteristics of the traffic are identified that might be suitable for classifying the traffic. These characteristics are referred to as features. Features can be associated with single packets, such as packet lengths, or associated with aggregated traffic statistics, such as means and standard deviations. Once the features have been identified the classifier is trained to associate particular features with a particular class of flows. Once trained, the classifier is tested on previously unseen flows. In our work, we are interested in classifying flows into Skype and non-Skype classes.

The process of constructing the classifier results in a set of rules or some other mechanism which can then be used to classify previously unseen traffic flows.

There are many different techniques for constructing classifiers. The broadest categories are unsupervised and supervised classifiers. Unsupervised classifiers attempt to group objects with similar features into clusters. This kind of classifier is commonly used in identifying different classes that may be present in a particular dataset but is less useful in classifying objects when the classes are already specified.

The other category of classifier is the supervised classifier. Supervised classifiers are presented with examples of object features and the class to which the object belongs. The classifier

adjusts its classification mechanism to best match the examples presented to it. It ‘learns’ that certain combinations of features are associated with certain classes. There are a number of different types of supervised classifiers. A rule-based classifier uses the training set to construct rules as to how objects presented to it in future should be classified. A common way of formulating rules is as a decision-tree, where a rule is applied and depending on the outcome other rules branching off that rule are applied until a classification outcome is achieved. A common way of constructing a decision tree is to use the C4.5 algorithm. We used the WEKA implementation of C4.5 in our work [14].

Common measures of the effectiveness of a classifier are recall and precision [12][14]. Recall is the percentage of samples that belong to that class that were correctly identified as such. Precision is the percentage of samples correctly classified as belonging to a particular class out of the total number classified as belonging to that class.

These values are calculated using the percentage of true-positives (the percentage of flows correctly identified as Skype traffic), true-negatives (the percentage of flows correctly identified as not Skype), false-positives (the percentage of flows incorrectly identified as Skype) and false-negatives (the percentage of flows incorrectly identified as not Skype).

If we denote true-positive percentage by TP , true-negative percentage by TN , false-positive percentage by FP , and false-negative percentage by FN , then we define recall and precision as follows:

$$recall = \frac{TP}{TP + FN} \quad (1)$$

$$precision = \frac{TP}{TP + FP} \quad (2)$$

It is worth noting that for a classifier to be effective it is important that both its recall and precision are high. A classifier which classified all samples presented to it as belonging to a particular class would have a very high recall, but a low precision.

4. EXPERIMENTAL METHOD

Our approach was to capture Skype and non-Skype traffic flows and segment the packets making up each flow into sliding windows of 1 to 10 seconds using 1 second increments. We then trained a classifier for each window size and evaluated its performance.

We used tcpdump to capture the Skype flows [15]. The software package Netmate was then used to process the PCAP files [17]. Netmate separates traffic into flows, determined by the usual tuple of protocol, source IP address, destination IP address and port number. We then used the R software package to do an initial analysis [18]. We then used the WEKA machine learning software package to construct and test the classifiers using data exported from Netmate.

Non-Skype traffic was obtained from two 24 hour traces from the University of Twente [16] and three hours of game traffic from Swinburne University of Technology. Our Skype traffic comprised 6.8 hours obtained from 18 calls made across multi-hop public networks. A total of 57 MB of Skype traffic was collected for analysis, comprising approximately 710,000 packets.

The trace from the University of Twente comprised approximately 503 Mbytes and approximately 752,000 flows.

As is standard practice in machine learning, we constructed two datasets, one for training and the other for testing. The training dataset comprised 3.3 hours of Skype traffic and one of the 24 hour traces from the University of Twente and 1.5 hours of game traffic from Swinburne University. The testing dataset comprised 3.5 hours of Skype traffic and the other 24 hour trace from the University of Twente and 1.5 hours of game traffic. We also used the common technique of cross validation. We trained the classifiers on one dataset and then tested it on the other and vice-versa. The results reported are the average of both tests.

Previously, most success in traffic classification has been obtained using Decision Tree classifiers and Naïve Bayes classifiers[12]. Our early experiments showed that most of the features of interest were not Normally distributed, meaning a Naïve Bayes classifier was unlikely to be effective. Consequently, after some initial experiments showing that this was indeed the case, we used the C4.5 Decision Tree Algorithm to construct our classifier.

5. RESULTS

5.1 Skype Characteristics

The statistics we found most effective in identifying Skype traffic were characteristic IP packet lengths, large packet statistics, and packet interarrival times. In this section we discuss the effectiveness of each class of characteristic in identifying Skype traffic flows and then consider the effectiveness of different combinations of them. Since there may be a huge number of flows to examine and classify, using as small a number of features as possible is desirable. Each feature requires some statistic or statistics of the traffic flow to be monitored, collected and possibly aggregated. However, a larger number of features are likely to result in a more accurate classifier. We examine the performance of classifiers using different numbers of features.

The first feature class we discuss are characteristic packet lengths. These are commonly occurring packet lengths that are less than 80 bytes. From a statistical analysis of packet lengths it appears that packet lengths less than 80 bytes are restricted to a limited number of values. These are 44, 45, 48, 51, 58, 60, 65, 73 and 74. They also occur frequently, making them a useful indicator of Skype traffic.

Another useful indicator of Skype traffic is the interarrival time between packets. The Skype Client generates packets at intervals that are a multiple of 16 milliseconds. We observed interarrival times of 16, 32, 48 or 64 milliseconds. A minor difficulty in using this as a feature was that transitions between these values occur frequently during Skype communication. To minimize the effects of this variability we experimented with two variants of interarrival times. The first was to include all interarrival times. The second was to only present windows to the classifier from a Skype flow where there were no transitions between interarrival times. We called this the dependent interarrival time. This second feature proved to be a very reliable indicator of Skype traffic.

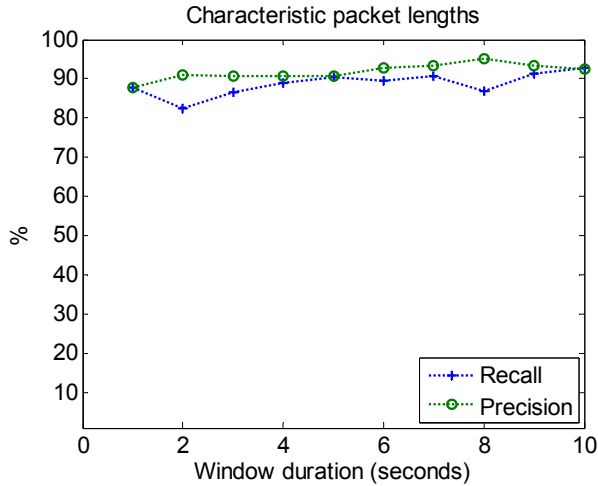


Figure 2. Recall and Precision using Characteristic Packet Lengths only

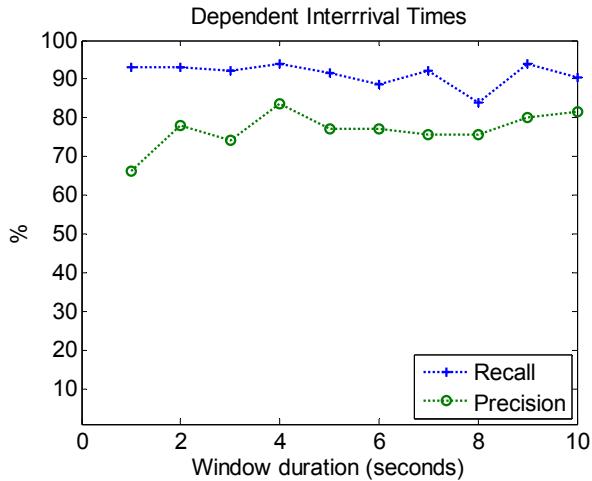


Figure 3. Recall and Precision using Dependent Interarrival Times only

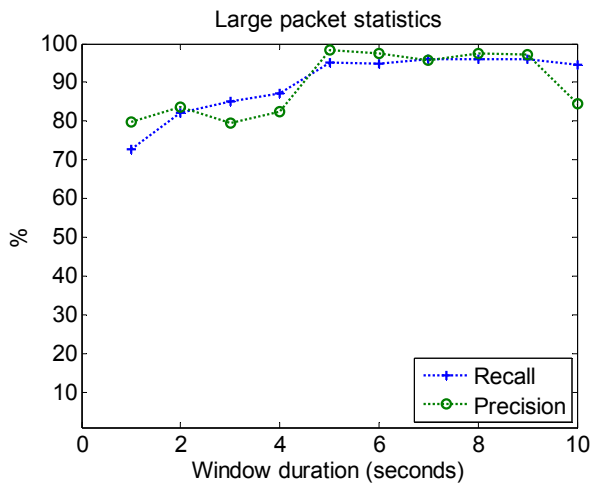


Figure 4. Recall and Precision using Large Packet Statistics Only

Large packet statistics are the final group of features we report on. In this feature class we used statistics across the whole window, rather than individual occurrences. These statistics were calculated on packets of length greater than or equal to 80 bytes. The statistics used were the minimum, maximum, median, lower-quartile, upper-quartile, mean and standard deviation.

5.2 Classification using one feature class

In this section we report on the performance of classifiers developed using just one of the feature classes described above.

We found characteristic packet lengths quite effective even on window sizes as small as 1 second. From Figure 2, we can see that window sizes of only 1 second provided recall and precision of 89%. The effectiveness of the classifier did not change greatly as the window length was increased remaining between 85 and 95% for window sizes up to 10 seconds.

The second feature we used was the packet interarrival time. This was less effective at identifying Skype traffic than the other two feature classes described in this section. Although providing good recall (mostly greater than 90%) Figure 3 shows that for window sizes up to 10 seconds, its precision was much lower, mostly between 70 and 80%.

The final feature class we report on in this section is the large packet length statistics. We found this to be a very effective feature for window lengths greater than 5 seconds. From Figure 4 we see that for window durations of 5 seconds or more the recall and precision are mostly in the mid to high 90s. However, for window sizes less than 5 seconds the statistics are much poorer with recall as low as 72%, precision as low as 75%.

Overall, large packet length statistics taken over periods of 5 seconds or more provide the best features for identifying Skype traffic flows, whereas characteristic packet lengths provide a quicker, but less effective measure. In the next section we consider using pairs of the classification classes.

5.3 Classification using two feature classes

We now consider different combinations of feature classes, taken two at a time. Since we may be monitoring many thousands of flows, ideally we would like to keep the number of measurements that need to be fed into our classifier as small as possible. In this section we show that we can improve on the performance of single measures, but at the expense of having to maintain additional information about each flow.

The performance of a classifier using characteristic packet lengths and interarrival times is shown in Figure 5. (Note that the y-axis shows percentages from 75 to 100). From Figure 5 we see that the performance of this classifier is superior to either of the features used in isolation. For window sizes of 1 second we obtain recall and precision of 94 and 92% respectively which is an improvement on recall and precision of interarrival time alone (92 and 65% respectively) and characteristic packet lengths alone (89% recall and precision). There is some slight improvement in performance as the window duration is increased.

Table 1 shows that short packet lengths are strongly correlated with specific interarrival times. In particular, when the interarrival time is 16 ms, 59% of packets less than 74 bytes are 51 bytes in length. When the interarrival time is 32 ms, 54.9% of packets are 44 bytes in length.

Table 1. Percentage of small packets and inter-arrival times

Length	44	45	48	51	58	60	65	73	74
Dataset	49.2	1.3	1.5	20.7	6.4	1.8	1.3	0.8	1.2
16 ms	28.7	0.7	1.3	59.0	1.2	0.4	0.1	0.5	1.5
32 ms	54.9	1.4	4.4	10.9	19.1	0.3	1.2	0.3	0.7
48 ms	36.6	0.9	3.6	1.6	33.5	0.1	11.2	0.1	0.1
64 ms	23.5	0.5	2.5	39.6	4.3	0.2	20.4	0.2	0.6

For an interarrival time of 48 ms, the percentage of packets 44 bytes in length is 36.6% and 58 bytes is 33.5%. This gives us some indication that combining both interarrival times and small packet lengths may give an improved classifier performance. We suspect this correlation is caused by the encoder reducing its packet rate during quiet times to minimize transport layer overhead.

The second pair of feature classes we consider are the large packet statistics and characteristic packet lengths. These are shown in Figure 6. Again, these features used together outperform the features used in isolation. For a window size of 1 second the recall is 93% and the precision is 96% which surpasses the recall and precision of the large packet lengths (recall 72% and precision 80%) and the Bidirectional Interpacket Arrival times (recall 95% and precision 66%). We see significant improvement in performance as the window duration increases to approximately 98 to 99% for window sizes greater than 5 seconds.

The final pair of features we consider are the large packet statistics and the interarrival times shown in Figure 7. This is the weakest classifier we describe in this section. For window sizes of 1 second we obtain recall and precision of 94 and 92% respectively which compares well with recall and precision of interarrival time alone (92 and 65%) and large packet lengths (recall 72% and precision 80%). However, there is considerable improvement in performance as the window duration is increased. For window sizes of 5 seconds or more recall is around 97%. Precision is between 90 and 95% for the same window sizes but declines to 85% for a 10 second window.

5.4 Classification using three feature classes

Finally we consider the effectiveness of all three of these features when used in a single classifier. Performance is shown in Figure 8. (Note the reduced y-axis scale). In this case the classifier performs well for window sizes of 5 or more seconds with precision mostly better than 99% and recall mostly better than 98%.

6. CONCLUSION

In this paper we reported on using a number of feature classes to classify IP flows as being Skype or non-Skype. Our goal was to classify Skype, using as short a duration as possible. The feature classes we found most effective were the interarrival times between IP packets, frequently occurring IP packet lengths less than 80 bytes and statistics of IP packets of length greater than or equal to 80 bytes. We found that using all three feature classes in a single classifier provided 98 percent precision and 99 percent recall when using a window duration of 5 seconds or more.

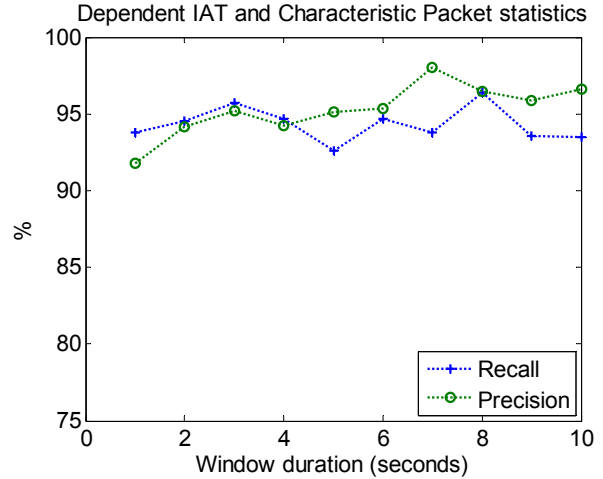


Figure 5. Recall and Precision using Dependent Interarrival Times and Characteristic Packet Lengths (Note reduced y-axis scale)

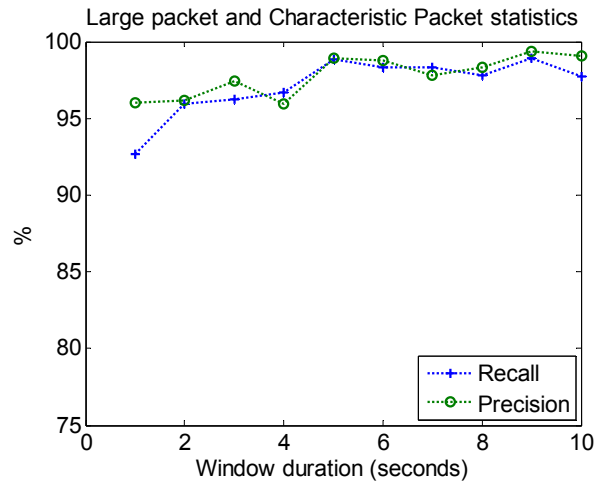


Figure 6. Recall and Precision using Large Packet Statistics and Characteristic Packet Lengths (Note reduced y-axis scale)

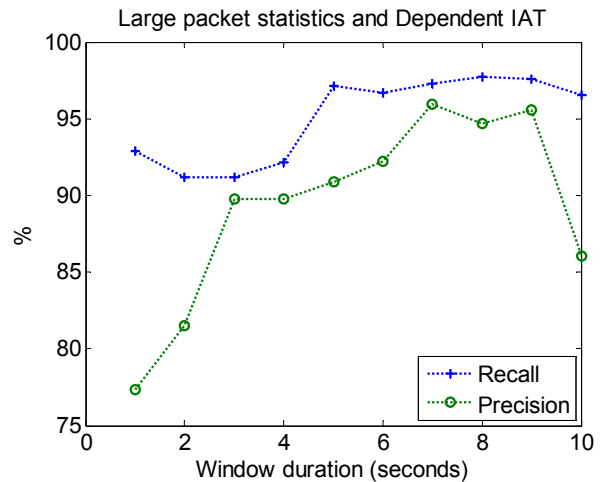


Figure 7. Recall and Precision using Dependent Interarrival Times and Large Packet Statistics (Note reduced y-axis scale)

This work has shown that it is possible to identify Skype traffic successfully using quite short samples of traffic. Our future work will be to identify additional features, particularly based on interarrival time, to identify Skype traffic using shorter windows than 5 seconds. We also plan to investigate whether these particular features generalize to other codecs used in the Skype application.

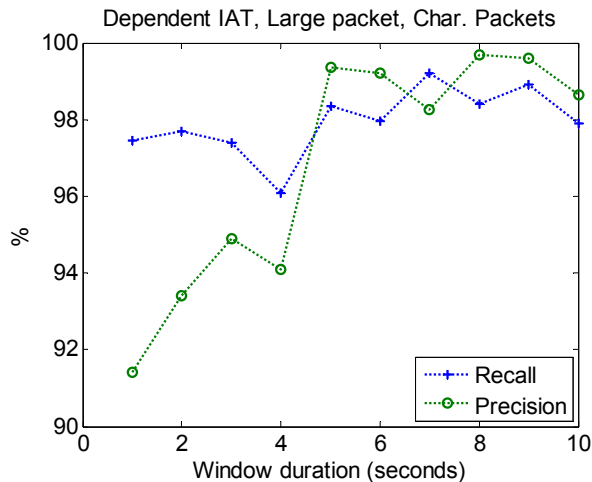


Figure 8. Recall and Precision using Interarrival Times, Large Packet Statistics and Characteristics Packet Lengths (Note reduced y-axis scale)

7. REFERENCES

- [1] Baset, S., Schulzrinne, H. 2006 An analysis of the Skype peer-to-peer Internet Telephony Protocol, IEEE INFOCOM'06.
- [2] Ebay results report, 3rd quarter 2008. <http://investor.ebay.com/results.cfm>, accessed 13 February 2009.
- [3] Skype website, <http://www.skype.com>, accessed 13 February 2009.
- [4] Bonfiglio, D., Mellia, M., Meo, M., Rossi, D., Tofanelli, P. 2007 Revealing Skype traffic: When Randomness Plays with you, ACM SIGCOMM'07.
- [5] Bonfiglio, D., Mellia, M., Meo, M., Rossi, D. 2009 Detailed Analysis of Skype Traffic, IEEE Transactions on Multimedia, vol 11, no1, pp. 117-127.
- [6] Suh, K., Figueredo, D., Kurose, J., Towsley, D. 2006 Characterising and Detecting Skype Relayed Traffic, IEEE INFOCOM'06.
- [7] CALEA Online, <http://www.calea.org>, accessed 13 February 2009.
- [8] Upson, S. 2007 Wiretapping Woes, IEEE Spectrum, May 2007.
- [9] Maloku, N., Aljaz, T., Dolenc, F. 2003 Legal Call Interception in Next Generation Networks, Proceedings of the 7th International Conference on Telecommunications.
- [10] Baker, F., Foster, B., Sharp, C. 2004 Internet Engineering Task Force, Cisco Architecture for Lawful Intercept in IP Networks, <http://www.ietf.org/rfc/rfc3924.txt>, Accessed 13 February 2009.
- [11] Bellovin, S., Blaze, M., Bricell, E., Brooks, C., Cerf, V., Diffie, W., Landau, S., Peterson, J., Treichler, J. 2006 Security Implications of Applying Communications Assistance to Law Enforcement Act to Voice over IP, Information Technology Association of America.
- [12] Nguyen, T. and Armitage, G. 2008 A Survey of Techniques for Internet Traffic Classification using Machine Learning, IEEE Communications Surveys & Tutorials, vol. 10 no. 4.
- [13] Lindblom, J. 2005 A sinusoidal voice over packet coder tailored for the frame-erasure channel, IEEE Transactions on Speech and Audio Processing, vol. 13, no. 5, pp. 787-798.
- [14] Witten, I., Frank, E. 2005 Data Mining: Practical Machine Learning Tools and Techniques, 2nd Ed, Elsevier Inc, San Francisco, CA.
- [15] Tcpdump, <http://www.tcpdump.org>, Accessed 13 February 2009.
- [16] University of Twente, Traffic Measurement Data Repository <http://traces.simpleweb.org/>, Accessed 13 February 2009.
- [17] Netmate, <http://www.ip-measurement.org/tools/netmate>, Accessed 13 February 2009.
- [18] R, The R Project for Statistical Computing, <http://www.r-project.org>, Accessed 13 February 2009.
- [19] Williams, N., Zander, S., Armitage, G. 2006 A Preliminary Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification, ACM SIGCOMM Computer Communication Review, vol. 36 no. 5 pp. 7-15.
- [20] But, J., Armitage, G., Stewart, L. 2008 Outsourcing Automated QoS Control of Home Routers for a Better Online Game Experience IEEE Communications, vol. 46 no. 12 pp. 64-70.
- [21] Nguyen, T., Armitage, G. 2006 Training on multiple sub-flows to optimise the use of Machine Learning classifiers in real-world IP networks in IEEE 31st Conference on Local Computer Networks, pp. 369-376. Tampa, Florida, USA.
- [22] Nguyen, T., Armitage, G. 2006 Synthetic Sub-flow Pairs for Timely and Stable IP Traffic Identification in Australian Telecommunication Networks and Application Conference 2006, pp. 293-297. Melbourne, Australia.