

CESNET Technical Report 6/2008

Is Spam Visible in Flow-level Statistics?

MARTIN ŽÁDNÍK, ZBYNĚK MICHLOVSKÝ

Abstract

This paper investigates feasibility of detection of spam connections using flow statistics collected upon SMTP connections only. To this end, the paper analyzes several days of SMTP communication collected at middle-sized email server. In order to prove that spam connections can be automatically identified at the TCP/IP layer we utilize supervised learning algorithm to construct classifier, in our case the decision tree. The quality of classifier is evaluated and results shows that the flow based statistics contain detectable fingerprint specific to spam connections. Such finding may help with further study of spam behavior in broader manner as the flow statistics can be collected on-line at the backbone links where it is possible to see SMTP traffic for more than one email server.

Keywords: network measurement, spam, identification, characteristics

1 Introduction

There are many methods and anti-spam techniques of filtering unsolicited mail. One of the most widely used and the most effective method is DNS blacklisting (DNSBL), which filters incoming mail on the basis of identifying spammer's IP address. Majority of mail servers is configured to reject or flag email that has been sent from IP that is listed in one of many DNSBL-databases. Another techniques are based on recognizing the pattern or regular expression in email messages. Other methods rely on strict adherence of RFC, e.g. helo/ehlo checking, graylisting, etc. Additionally, very popular are filters that include statistical methods like Bayes filters [6] or elements of artificial intelligence like neural networks. Less common approaches for filtering spam are based on comparison between CRC of receiving email and a database of CRC's junk messages, such as Distributed Checksum Clearinghouses¹.

Popular way of spreading spam is to use botnets. It is a group of remotely controlled computers compromised by a hacker, computer virus or Trojan. Bots do not send spam in bulks, which makes it difficult for Internet service providers (ISP) to separate spam-traffic from regular traffic at a glance. On the other hand ISPs are able to filter all outgoing SMTP communication coming from open relay servers and thus eliminate the possibility of being source of spam themselves unfortunately it is rarely used. Another approach of filtering unsolicited emails coming from spam relay machines is described in [7]. Authors suggest detecting excessive number of SMTP connections established by hosts on the monitored network segment. Spam filtering method presented in [8] is based on classification of mail delivery traffic into different categories by the similarity of messages contents. If the number of similar mails in any category exceeds a spam threshold, then this category is marked

¹ <http://www.rhyolite.com/anti-spam/dcc/>

as a spam one. In contrast with these methods, our approach is focused on identifying SPAM-flow directly on the network layer.

In similar way, authors of [5] study the network-level behavior of the spam traffic, including: IP address ranges that send the most spam, common spamming modes (e.g. BGP route hijacking, bots), how persistent across time each spamming host is, and characteristics of spamming botnets. But they do not aspire to anatomize the flow characteristics to the depth.

Our work is inspired by successful utilization of traffic flow characteristics to classify network traffic into categories [3] or to reveal traffic of specific application, for example to detect VoIP traffic [1]. In this context, we presume that spammers utilize dedicated SMTP engines to spread spam and these engines differ from other SMTP implementations. Therefore it should be possible observe them in statistics such as TCP window size or packet size, inter-packet intervals, etc.

This paper contributes to the current research of spam filtering and analysis of spam behavior. It provides the evidence that it is possible to detect spam based on statistical tracking TCP/IP flow characteristics rather than to inspect content of the connection itself. The goal of presented approach is not to filter 100% of junk email messages. Its main contribution is in spam detection at the level of network backbone traffic where it is possible to observe behavior of spamming hosts in broader context as well as provide additional information in order to support spam-blocking techniques.

The structure of the text is organized as follows: first, in section 2 we present the setup of traffic measurement and annotation architecture. Next, section 3 gives a short overview about decision tree classifier used for automated classification of SMTP traffic. Collected data and results of classification are analyzed in section 4. Finally, section 5 briefly summarizes the paper and outlines our future work.

2 Experimental Setup

We collected data from the SMTP server hosting mailboxes the Liberouter² project group. The online service DNS-Blacklisting was switched on thus allowed us to obtain the part of SMTP flow where the connection from blacklisted IP is refused. We suppose the majority of SMTP servers use DNS-Blacklisting as an effective counter measure against spam. Emails that made it through the DNS-Blacklisting were delivered into users' mailboxes and at the same time were stored for further offline analysis.

In parallel, all SMTP and SMTPS traffic was dumped in the file. The file was processed later on by a script that measured 30 unidirectional flow characteristics in each direction per each connection. We have chosen only characteristics that are potentially feasible to measure online over high-speed network traffic. Following constraints were applied:

1. one pass through data set to enumerate the characteristic
2. small amount of memory (max. 8B) to store the item characteristic during the measurement of the flow.

² <http://www.liberouter.org>

For example maximum, minimum, average, variance were traced for length of packet, interval between consequent packets of the same flow, TCP window size and other values (a subset of discriminators presented by Moore in [4]).

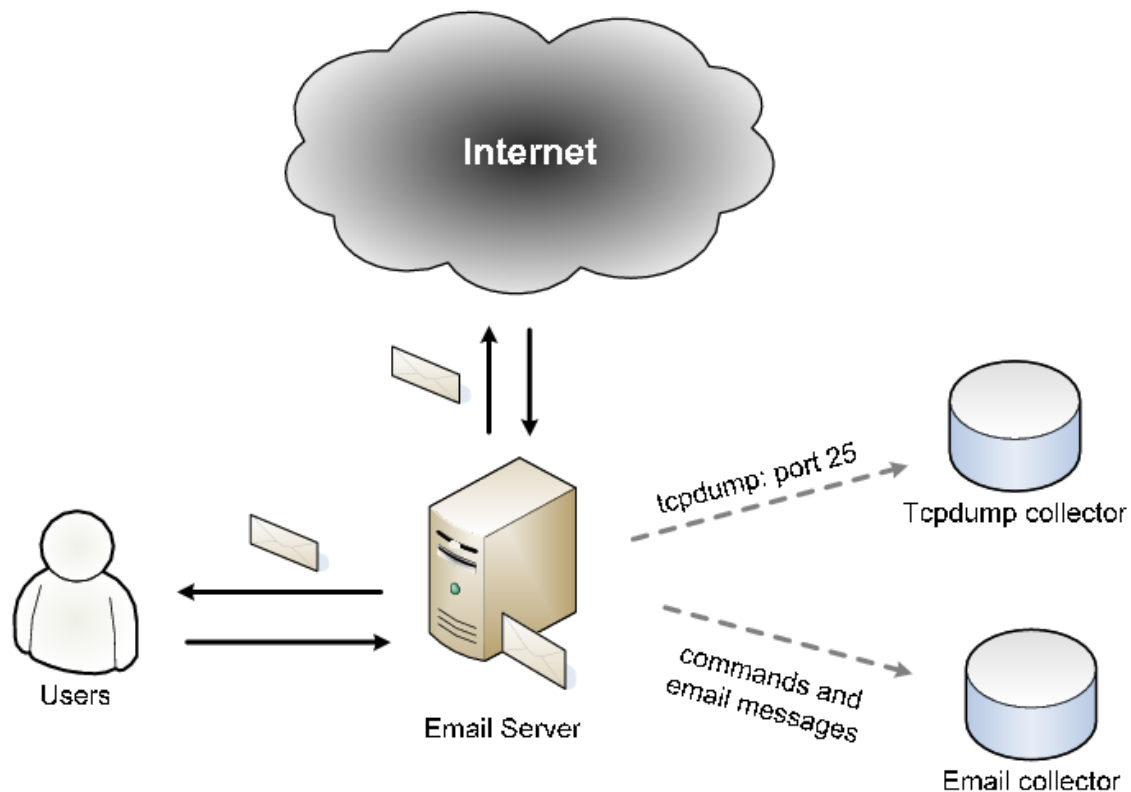


Figure 1. Experimental setup.

In offline mode, the delivered mails were classified by SpamAssassin³ version 3.2.3 into two groups: relevant emails and spam. For mail filtering were used these components: Bayes System (with auto-learning); Network test as a RBL checks, Razor2, DCC and Pyzor; etc. Czech and English were set as accepted languages. Additionally, we left threshold score on a default value 5.0. Besides, we applied the configuration tool SpamAssassin Configuration⁴ for the basic settings and then adjusted the configuration manually (e.g., learning Bayes). After the classification of these emails, it was possible to assign a label to each flow which specified the type of the network communication. It could be classified as a relevant email, spam or other communication (attack, outgoing email, failed communication). By using these actions and data modification we obtained data that were ready for classification 2.

³ <http://spamassassin.apache.org/>

⁴ <http://www.yrex.com/spam/spamconfig.php>

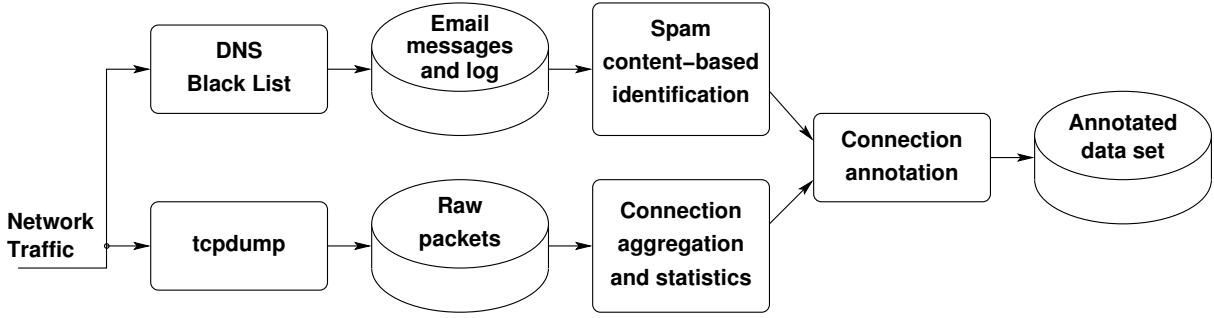


Figure 2. Analysis pipeline.

3 Classification

Our intention is to use suitable classification method to validate that spam can be detected in statistics collected upon SMTP connection. The classification method should satisfy following criteria:

1. Low cost of evaluation
2. Accurate results
3. Learning process including the feature selection
4. Easy to understand

The key characteristics of every classification method is high accuracy but at the same time the trained classifier must be simple and easy to evaluate. Fast evaluation is of the great importance considering it should run in the device processing multi-gigabit backbone link. On the other hand the training process may be quite complex and it may take several hours to train the classifier.

The essential task is to select proper features from the feature set which contribute to the some classification methods can deal with feature selection during the learning process, some methods require to have the feature set filtered otherwise the training process does not work well.

Our goal is not to tune the classification but rather to prove its feasibility for spam detection. Therefore we would like to use more sophisticated method to train the classifier that can deal with feature selection as well. We have chosen the decision tree induction approach as best fitting to our requirements.

The decision tree is build to recursively partition the data into smaller subsets until residual instances in the leaves belong to the same class. Following algorithm describes the process of constructing the tree as described in [2].

- Initialize by setting variable T to be the training set.
- Apply the following steps to T .
 1. If all elements in T are of class c_j , create a c_j node and halt.
 2. Otherwise select a feature F with values v_1, v_2, \dots, v_N . Partition T into T_1, T_2, \dots, T_N , according to their values on F . Create a branch with F as a parent node and T_1, T_2, \dots, T_N as child nodes.
 3. Apply the procedure recursively to each child node.

The feature selection is a fundamental part of decision tree induction. It is based on selecting feature F that minimizes the information $I_F(S)$ necessary to further classify subsets received by partitioning of set S using feature F . In other words,

to maximize the information gain $G(F)$ which is computed as a difference of the entropy of set S and the entropy after partitioning by F .

$$G(F) = I(S) - I_F(S),$$

$$I(S) = - \sum_{i=1}^n p_i \log_2 p_i,$$

$$I_F(S) = - \sum_{j=1}^v \frac{|S_j|}{|S|} I(S_j).$$

Such approach allows to choose good features and create simple but not necessarily the optimal tree.

During our experiments we used Weka⁵ that implements several modifications of decision tree induction algorithm. It is well known that each classification algorithm or its modification may perform differently for particular problems. Therefore tests were conducted with available decision tree algorithms and results of the best performing are presented in Section 4.2. In our case Random Forest performed the best. It constructs several decision trees and the class of the instance is determined by the most votes over all the trees.

4 Results

Results of our measurement are twofold. General properties of measured data are discussed first and afterward feasibility of flow based statistics for task of spam detection is analyzed.

4.1 Measured Data

The data were collected during 9 days from 07/04/2008 to 16/04/2008 at the email server. The server hosts over one hundred mailboxes for participants of Liberouter project. Participants are advised to use these addresses carefully, e.g. do not use them as a contact emails to register for free service etc.

The raw communication, i.e. raw packets, was dumped with usage of tcpdump. The total size of the file was 235 MB after nine days.

In parallel to tcpdump, received emails were saved to a separate folders as described in Section 2. The total number of received messages was 11559 with the size of 278 MB. The log of SMTP communication took also about 270 MB.

Collected messages were annotated by the parsing script which used log of SMTP communication and the results of SpamAssassin. The annotated dataset was manually and randomly checked for correctness; about 2% of emails were annotated incorrectly (mostly weekly reports were identified as spam but it was easy to correct them, the rest was spam that seemed to have a more serious content and was classified as non-spam, i.e., y_spam).

The first point of our interest was to find out whether one spamming host has sent more than one spam to our email server. Surprisingly, most of the spam servers

⁵ <http://www.cs.waikato.ac.nz/ml/weka/>

have sent only one spam during the whole measurement period. The average number of accepted spams per host is 1.30 and the average number of rejected spams per host is 1.71 (histogram is displayed in Figure 3)

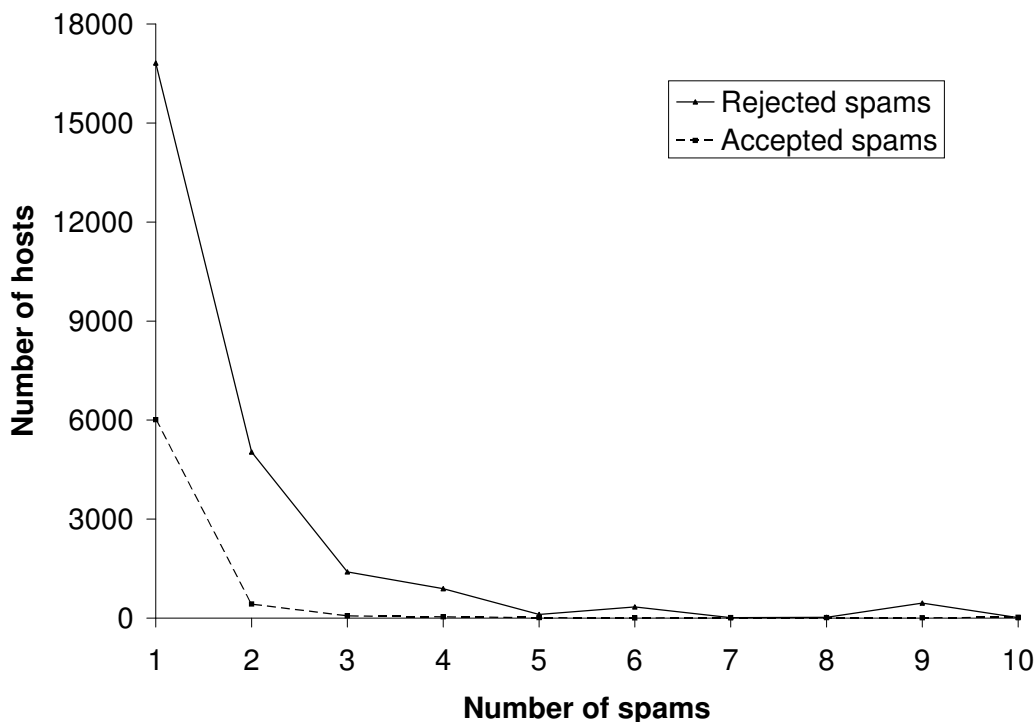


Figure 3. Histogram of spam sources in dependence on the amount of spam

Figure 4 outlines the obtained number of spams in dependence on IP addresses. It seems that from certain IP ranges like 94.X.X.X to 115.X.X.X there are no received spams at all. The reason might be that these subnets allow to send emails via dedicated gateways and so prevent themselves from being a source of spam. It is important to mention that the source IP addresses cannot be spoofed due to the bidirectional communication required by SMTP for successful message transfer and delivery.

Further, we wanted to find out if the amount of received spam is correlated in the time domain, i.e., if the amount of spam forms a pattern that is repeated every day or for any longer period. Because our measurement lasted only for nine days, the longest visible period may be only shorter. The amount of received spam per hour was aggregated and plotted in the graph in Figure 5. One significant outlier emerged on the 14th of April at 6 a.m. with 268 accepted spams. The detail analysis revealed that our colleague became a victim of email address spoofing, i.e. spammers used his email address and inserted it in the field "From:" in their spam messages. Most of the target email servers recognized such emails as a spam and blocked it. At the same time they replied to our college with emails containing the original spam and the message about refusal. Consequently, these emails were recognized correctly as unsolicited email during the annotation process. Such indirect

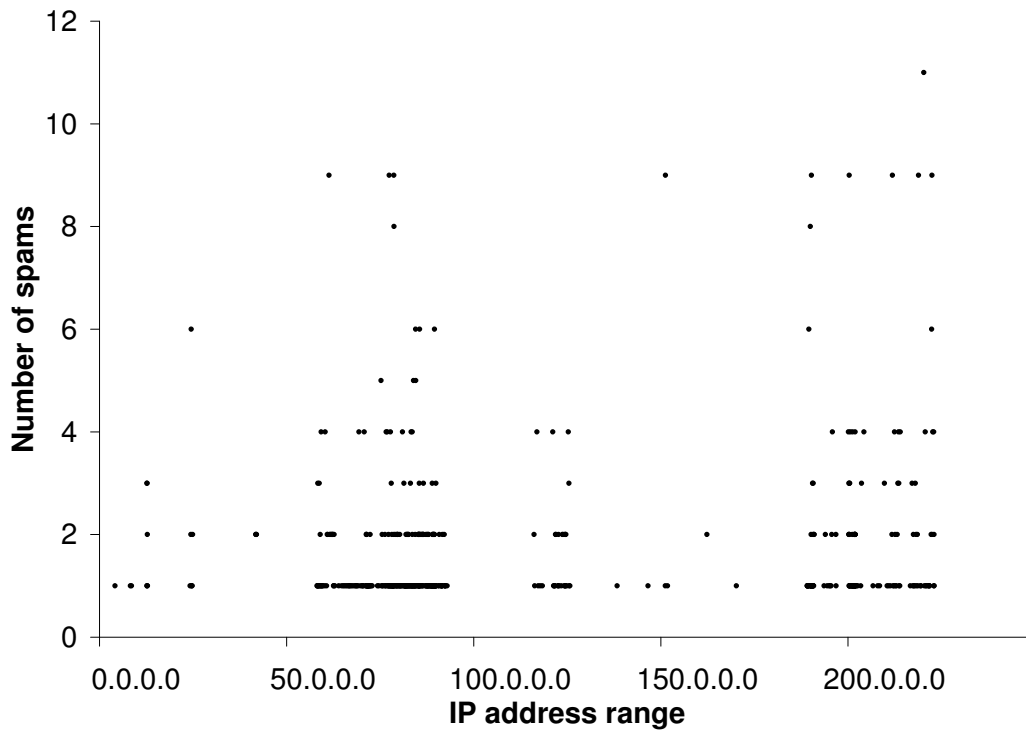


Figure 4. Distribution of spam in dependence on IP addresses

spam was not in the scope of our time analysis and is omitted otherwise it would corrupt the frequency characteristic because of small traffic sample.

Discrete Fourier transform was used to reveal any dominant frequency in the signal. The only observable coefficient was equivalent to one day period. Other periods were not as dominant or were hidden because of short measurement interval.

The interesting behavior of the 24-hour period is that the minimum of daily received spams is located around the midnight and the maximum around the noon. One can only presume that spammers utilize local zombies that transmit spam when switched on and therefore the characteristics follow the time zone of the victim ($GMT+1$ in our case) and as a consequence the daily distribution of spam is similar to legitimate emails (y_{spam}). Further investigation of this behavior is necessary but is out of the scope of this paper.

4.2 Results of Classification

In this section, we experiment with several configurations of training set to explore characteristics within the context of classification efficiency. First experiment investigates the fundamental ability of classifier to distinguish among several classes of SMTP traffic, namely:

1. *ham* – consists of connections that have been successfully received and marked as not spam by SpamAssassin
2. y_{spam} – connections that have been successfully received and marked as spam by SpamAssassin
3. *rejected* – connections that have been rejected because of DNS black list

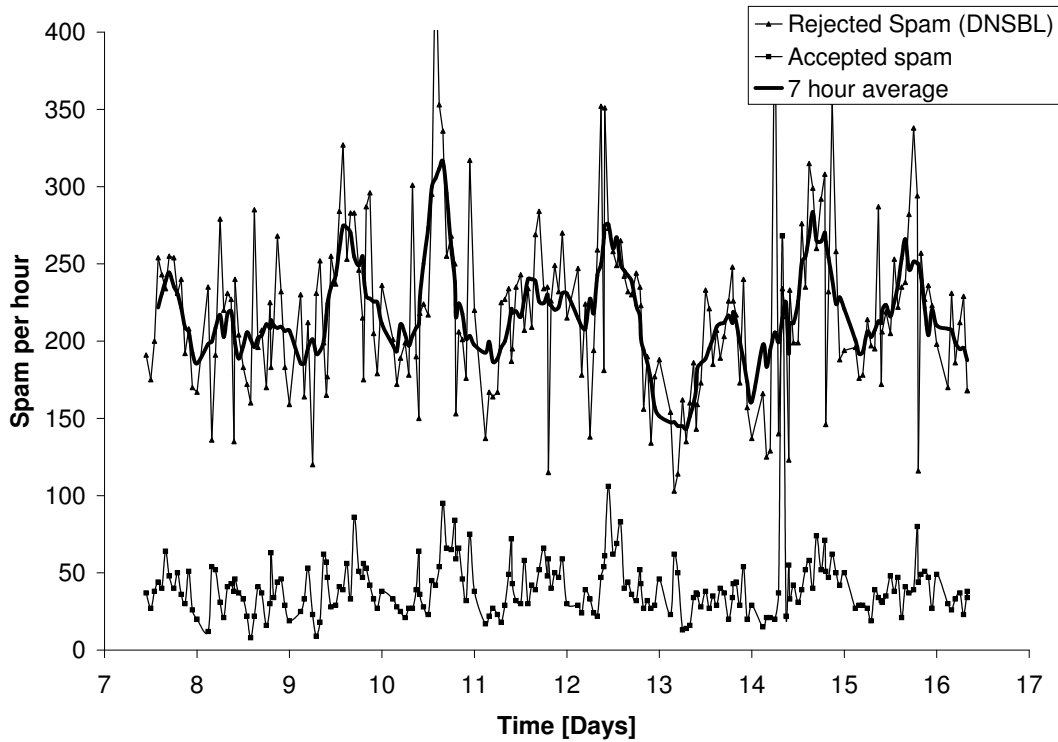


Figure 5. The amount of spam received per hour plotted for the whole measurement period (9 days).

4. *outgoing* – connections that originate from our server in order to transfer messages to other email server
5. *other* – traffic caused by scanning, DoS, etc.

Despite connections marked as *y_spam* and *rejected* constitute spams, it makes sense to assign them into two different classes as it is expected that their connections behave differently and have different fingerprints in network traffic statistics. The distribution of connections into classes is denoted in the Tab. 1.

Table 1. Distribution of annotated SMTP connections into classes.

<i>y_spam</i>	<i>ham</i>	<i>rejected</i>	<i>outgoing</i>	<i>other</i>	total amount
11222	1554	38314	2618	4334	58052

In our first experiment, training set consists of 66% of all annotated traffic statistics and the remainder serves for evaluation of the classifier accuracy. The training of the classification model took approximately 4 minutes while the evaluation was very fast (less than 1 s). Results confirms that the classification of SMTP traffic into specified categories is possible, moreover quite a small error rate can be achieved (2.1% of incorrectly classified instances). Closer details are presented in Tab. 2 where each column denotes how many instances of given category were correctly and incorrectly classified.

The outgoing emails are easily recognized due to the opposite direction of these connections to the remainder. Likewise, the classifier easily recognizes connections

that were blocked by DNSBL. On the other hand its performance to distinguish between accepted spams (*y_spam*) and legitimate emails *ham* is not as bright. If the classifier is trained only to separate these two classes then the results are slightly better but still it would not be wise to detect and block potential spams using this approach. Higher error rate could be caused by similar characteristics of connections, small traffic sample or inaccurate annotation.

Table 2. Confusion matrix of classifier evaluated on 33% of all connections.

Classified as	<i>y_spam</i>	<i>ham</i>	<i>rejected</i>	<i>outgoing</i>	<i>other</i>
<i>y_spam</i>	3587	24	76	0	56
<i>ham</i>	193	540	2	0	6
<i>rejected</i>	23	0	12949	0	99
<i>outgoing</i>	0	0	0	896	0
<i>other</i>	44	4	135	0	1276
Precision	93.2%	95.5%	98.4%	100%	88.1%

The goal of our second experiment was to find out whether the classifier would still work if it is trained and evaluated only on the incoming flow of the connection. The experiment is motivated by situations where it is possible to monitor only one direction of the connection, for example due to asymmetric routing. The obtained results are given in Tab. 3. The performance of classifier that observes the incoming stream only is nearly the same as of the bidirectional one.

Table 3. Confusion matrix of classifier for incoming direction.

Classified as	<i>y_spam</i>	<i>ham</i>	<i>rejected</i>	<i>outgoing</i>	<i>other</i>
<i>y_spam</i>	3546	28	138	0	48
<i>ham</i>	189	531	4	0	5
<i>rejected</i>	7	0	12986	0	78
<i>outgoing</i>	0	0	0	896	0
<i>other</i>	60	4	181	0	1214
Precision	93.3%	90.3%	97.6%	100%	90.3%

The aging of the classifier was point of our interest during the last experiment. The classifier trained on whole measurement period was evaluated on data (one week) collected two months later. Performance of both classifiers, bidirectional and incoming flow, degraded significantly. The classifier failed to distinguish between spam and legitimate emails. The total classification error was about 6.1%, closer details are given in Tab. 4.

In order to keep the classifier accuracy high, it should be regularly retrained, we suggest on weekly basis according to the following experiment. The classifier was trained on data collected during one day (8th) and evaluated on the rest of the days in sequence. The ability to correctly distinguish between spam and legitimate emails is not significantly influenced during one week (the average error rate is about 7%).

Table 4. Confusion matrix of classifier evaluated 2 months later.

Classified as	y_spam	ham	rejected	outgoing	other
y_spam	9683	941	279	0	319
ham	799	675	15	45	20
rejected	26	0	37860	0	428
outgoing	0	0	3	2602	13
other	162	23	525	0	3624
Precision	90.1%	41.3%	97.9%	98.3%	82.5%

5 Conclusion

The paper presented an alternative approach to spam detection. In comparison to other popular approaches based on content analysis our method utilizes only TCP/IP flow characteristics to reveal spam connections. Measured characteristics were used to train and evaluate the performance of decision tree classifier. Results of our experiments shows that the spam connections are distinguishable to other SMTP communication.

In near future, we plan to train the classifier on a much larger training set to increase its accuracy. Consequently we intent to use it at the backbone link to classify traffic for more SMTP servers. If it works well we would like test it as an input for DNSBL technique to find out the feasibility of such an approach to spam elimination. Other experiments will be focused on measurement of spamservers such as ratio for different target email servers and also on behavior of spamming hosts, i.e., that spam follows local timezone of the victim.

References

- [1] BONFIGLIO, D.; MELLIA, M.; MEO, M.; ROSSI, D.; TOFANELLI, P. Revealing skype traffic: when randomness plays with you. *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, p. 37–48, 2007.
- [2] LIU, H.; MOTODA, H. *Feature Selection for Knowledge Discovery and Data Mining*. Kluwer Academic Publishers: Norwell, MA, 1998.
- [3] MOORE, A. W.; ZUEV, D. Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*. ACM: New York, 2005, pp. 50–60.
- [4] MOORE, A.; ZUEV, D.; CROGAN, M. *Discriminators for use in flow-based classification*. Technical Report RR-05-13, Department of Computer Science, Queen Mary University of London, 2005.
- [5] RAMACHANDRAN, A.; FEAMSTER, N. Understanding the network-level behavior of spammers. In *Proc. SIGCOMM 06*, September 11–16, Pisa, Italy, 2006.
- [6] SAHAMI, M.; DUMAIS, S.; HORVITZ, E. A bayesian approach to filtering junk e-mail. *AAAI Workshop on Learning for Text Categorization*, July 1998, Madison, Wisconsin. AAAI Technical Report WS-98-05.

- [7] SANDFORD, P.J.; SANDFORD, J.M.; PARISH, D.J. Analysis of smtp connection characteristics for detecting spam relays. In *Proc. International Multi-Conference on Computing in the Global Information Technology*, 2006, p. 68. ISBN 0-7695-2629-2.
- [8] ZHANG, N.; JIANG, Y.; FANG, B.; CHENG, X.; GUO, L. Traffic classification-based spam filter. In *IEEE International Conference on Communications*, vol.~5, p. 2130–2135, 2006.