DETECTION OF NETWORK ANOMALIES USING RANK TESTS

Céline Lévy-Leduc

CNRS/LTCI/Télécom ParisTech
37/39, Rue Dareau - 75014 Paris - Email: celine.levy-leduc@telecom-paristech.fr

ABSTRACT

We propose a novel and efficient method for on-line detection of network anomalies that lead to changes in Internet traffic such as (distributed) denial-of-service ((D)DoS) attacks. Our method consists in a data reduction stage based on record filtering followed by a nonparametric change-point detection test based on U-statistics. With such a method, we can address massive data streams and provide an on-line anomaly detection as well as the source and destination IP addresses involved. We apply this algorithm to some Internet traffic generated by France-Télécom Internet Service Provider (ISP) in the framework of the ANR-RNRT OSCAR project. This approach called *TopRank* in the following is very attractive since it enjoys a low computational cost and is able to detect several types of anomalies such as TCP/SYN flooding, UDP flooding, PortScan and NetScan with a low false alarm rate.

1. INTRODUCTION

Recent attacks on very popular web sites such as Yahoo, eBay and CNN leading to a disruption of services to users have triggered an increasing interest for network intrusion detection. Typical examples include Denial of Service (DoS) attacks – a network-based attack in which agents intentionally saturate system resources – their distributed version (DDoS), worm-based attacks and Address Resolution Protocol (ARP) Man In the Middle (MIM) attacks. Since the aforementioned attacks represent serious threats for computer networks, finding tools such as Intrusion Detection Systems (IDS) for ensuring the defense against them has become a major concern.

Existing IDS to deal with DoS attacks such as TCP (Transmission Control Protocol) SYN flooding, UDP flooding, PortScan and NetScan are based on two different approaches.

The first one is a signature-based approach which compares the observed patterns of the network traffic with known attack templates. If the attack belongs to the set of known attacks listed in the database then it can be successfully detected: Bro [12] and Snort [13] are two examples of such IDS. The obvious limitation of such an approach is the requirement that the signature of the anomaly has to be known in advance.

The second one is based on statistical tools which do not require any prior information about the kind of anomalies we are faced with. As a consequence, this approach can detect anomalies which do not belong to a prescribed database. A training stage is nevertheless required in order to learn the characteristics of legitimate traffic. Indeed, an alarm will be launched each time a deviation in the parameters of the studied traffic is observed with respect to the learned legitimate traffic. The latter approach uses the fact that anomalies in the network traffic lead to abrupt changes in some observations which have to be chosen according to the type of attacks (DoS, worm, MIM,...) we are looking for. These changes occur at unknown time instants and have to be detected as soon as possible. Detecting an attack in the network traffic can thus be described as a change-point detection problem which is a classical issue in statistics. The detection can either be performed with a fixed delay (batch approach) or with a minimal average delay (sequential approach). We refer to [1, 2, 4] and the references therein for a complete overview of the existing methods in statistical change-point detection.

The most widespread change-point detection technique in the field of network anomaly detection is the cumulated sum (CUSUM) algorithm which was first proposed by Page in [11]. The CUSUM algorithm has already been used by [17] and [14] for detecting DoS attacks of TCP/SYN flooding type. Such an attack consists in exploiting the TCP's three-way hand-shake mechanism and its limitation in maintaining half-open connections. More precisely, when a server receives a SYN request, it returns a SYN/ACK packet to the client. Until the SYN/ACK packet is acknowledged by the client, the connection remains half-opened for a period of up to the TCP connection timeout. A backlog queue is built in the system memory of the server to maintain all half-open connections, thus leading to a saturation of the server. In [17] and [14], the authors use the CUSUM algorithm to look for a change-point in the time series corresponding to the sum of received SYN packets and to the difference of the number of SYN and FIN packets respectively. With such an approach it is only possible to raise an alarm when a change occurs in the aggregated series but it is impossible to pick out the malicious flow. In [15] and [16], a multichannel detection procedure is proposed. This is a refined version of the algorithm previously used: it detects changes which occur in a channel and which could be obscured by the normal traffic in the other channels if global statistics were used.

Operators seeking to understand and manage their networks are increasingly looking at network-wide traffic flows using tools like Netflow [3]. Since each flow is characterized by 5 fields: source and destination IP addresses, source and destination ports and protocol number, a database of size up to 2^{104} has to be stored and studied. In order to detect anomalies over time of such massive data streams, dimension reduction techniques have to be used. Two main approaches have been proposed: random aggregation of IP flows (sketches) [7], [9] and PCA (Principal Component Analysis) techniques [8].

In this paper, we propose an intrusion detection method for identifying DoS and DDoS attacks in Internet traffic such as: TCP/SYN flooding, UDP flooding, PortScan and NetScan. Recall that UDP flooding is an attack similar to SYN flooding which aims at saturating the memory of a des-

tination IP address by sending a lot of UDP packets. The PortScan consists in sending TCP packets to each port of a machine to know which ones are opened. For the NetScan, a source IP address sends packets to a group of machines. Since we aim here at addressing massive data streams, the first stage of our approach is a data reduction step. It is then followed by a statistical batch nonparametric change-point test using rank statistics. A record filtering method is used as a reduction stage in the algorithm *TopRank* that we propose. This method is thoroughly described in Section 2. The corresponding algorithm has been implemented in C and applied to real datasets corresponding to some Internet traffic provided by France-Télécom within the framework of the ANR-RNRT OSCAR project to detect network anomalies of the previous types. The results are reported in Section 3. The method that we propose can be used to analyze a large amount of data and to provide an on-line anomaly detection algorithm as well as the source and destination IP addresses involved.

2. DESCRIPTION OF THE TOPRANK ALGORITHM

In the following, we describe the *TopRank* algorithm which can detect different types of DoS attacks such as TCP/SYN flooding, UDP flooding, PortScan and NetScan on-line. The raw data used by the *TopRank* algorithm consist in Netflow type data collected at several points of the Internet network. They include the source and destination IP addresses, the source and destination ports, the start time and the end time of the flow as well as the protocol and the number of exchanged packets. Depending on the type of the attack, some time indexed traffic characteristics are of particular interest and have to be processed for detection purposes. For instance, in the case of the TCP/SYN flooding, the quantity of interest is the number of TCP/SYN packets received by each destination IP address per unit of time.

In the following, we propose to use a batch approach. More precisely, we analyze the traffic in successive observation windows each having a duration of $P \times \Delta$ seconds. We take a decision concerning the presence of potentially attacked IP addresses at the end of each observation window and identify the IP addresses involved. The parameter Δ corresponds to the smallest time unit used for building time series from Netflow type data. The integer P is the length of the time series within each window. More precisely, the time series are built as follows: in the case of the TCP/SYN flooding for instance, we shall denote by $N_i^\Delta(t)$ the number of TCP/SYN packets received by the destination IP address i in the sub-interval t of size Δ seconds. The corresponding time series of the destination IP address i will thus be $\left(N_i^\Delta(t)\right)_{1\leq t\leq P}$. In the case of UDP flooding, $N_i^\Delta(t)$ will be the number of UDP packets received by the destination IP address i in the sub-interval t of size Δ seconds. For the PortScan, we shall take as $N_i^\Delta(t)$ the number of different requested destination ports of the destination IP address i in the sub-interval t of size Δ seconds and for the NetScan, it will be the number of different requested destination IP addresses by the source IP address i.

A crude solution for detecting such kinds of anomalies consists in finding a change in the time series $\left(N_i^\Delta(t)\right)_{1\leq t\leq P}$ for all the possible IP addresses i encountered in each observation window of size $P \times \Delta$ seconds. Since the number of IP addresses can be huge, up to 2^{32}, we are faced in practice with massive data streams implying a construction and an analysis of several thousands of time series even for short observation periods (around 1 minute). To overcome this difficulty, a data reduction stage must precede the change-point detection stage. The data reduction stage is performed in the *TopRank* algorithm using a record filtering which will be further described below. As for the change-point detection step, we used nonparametric rank tests which do not require any prior information concerning the distribution of the observed traffic.

More precisely, the *TopRank* algorithm can be split into three steps described hereafter. Note that the following processing is performed in each observation window of length $P \times \Delta$ seconds and that all the stored data are cleaned up at the end of each observation window. Note also that we only describe the algorithm for the TCP/SYN flooding since the adaptation to the other kinds of attacks previously quoted is straightforward given the previous remarks.

Step 1: Record filtering

In each sub-interval of duration Δ seconds of the observation window, we select, for instance in the case of TCP/SYN flooding, the M destination IP addresses which have received the largest number of TCP/SYN packets. The indexes of these destination IP addresses are rearranged in such a way that: $N_{i_1}^\Delta(t) \geq N_{i_2}^\Delta(t) \geq \cdots \geq N_{i_M}^\Delta(t)$. In the following, $T_M^\Delta(t) = \{i_1, \ldots, i_M\}$. In other words, for all $t \in \{1, ..., P\}$, $\#T_M^\Delta(t) = M$ and for all $i \in T_M^\Delta(t)$ and $j \notin T_M^\Delta(t)$, $N_i^\Delta(t) \geq N_j^\Delta(t)$.

Step 2: Creation of censored time series

In this stage, we shall only focus on the destination IP addresses which belong, at least once in the observation window, to the set of the M' IP addresses which have received the largest number of TCP/SYN packets, where $1 \leq M' \leq M$. This means that we shall construct censored time series only for these destination IP addresses. More formally, we shall only focus on the IP addresses i belonging to

$$I = \bigcup_{t=1}^{P} T_{M'}^\Delta(t) \, .$$

Then, the corresponding time series are built as follows. For each $i \in I$, the value of the time series in the sub-interval t of length Δ seconds will be denoted by $X_i^\Delta(t)$ and will be defined in the following way. The value of $X_i^\Delta(t)$ will be equal to the number of TCP/SYN packets received by the destination IP address i if this destination IP address belongs to the set $T_M^\Delta(t)$. Otherwise $X_i^\Delta(t)$ will be equal to the number of TCP/SYN packets received by the M-th most requested destination IP address of the sub-interval t of size Δ. By construction, the time series are censored. More formally, for each destination IP address $i \in I$, the corresponding observations are:

$$(Y_i^\Delta(t))_{1\leq t\leq P} = (X_i^\Delta(t), \delta_i^\Delta(t))_{1\leq t\leq P}$$

where, for each $t \in \{1,...,P\}$,

$$X_i^\Delta(t) = \begin{cases} N_i^\Delta(t), & \text{if } i \in T_M^\Delta(t) \\ \text{Min}_{j \in T_M^\Delta(t)} N_j(t), & \text{otherwise,} \end{cases}$$

$$\delta_i^\Delta(t) = \begin{cases} 1, & \text{if } i \in T_M^\Delta(t) \\ 0, & \text{otherwise.} \end{cases}$$

The value of $\delta_i^\Delta(t)$ tells us if the corresponding value $X_i^\Delta(t)$ has been censored or not. Observe that, by definition, $\delta_i^\Delta(t) = 1$ implies $X_i^\Delta(t) = N_i^\Delta(t)$ and $\delta_i^\Delta(t) = 0$ implies $X_i^\Delta(t) \geq N_i^\Delta(t)$.

Step 3: Change-point detection test

In [6], a nonparametric statistical change-point detection method is proposed to analyze censored data as well as a way of computing its p-values. It is a nonparametric rank test using a score function (denoted by A in the following) which was first introduced by [5] and [10] in their generalization of Wilcoxon's rank test for censored data. We apply this test to each time series created in Step 2. Note that each of these time series is removed when the analysis in a given observation window is complete. With such an approach, up to $M' \times P$ time series of length P are processed in each observation window of time length $P \times \Delta$ seconds.

Let us now describe further the statistical test that we perform. This procedure aims at testing from the observations $(Y_i^\Delta(t))_{1 \leq t \leq P} = (X_i^\Delta(t), \delta_i^\Delta(t))_{1 \leq t \leq P}$ if a change occurred in the time series $\left(N_i^\Delta(t)\right)_{1 \leq t \leq P}$ for a given $i \in I$. More precisely, if we drop the dependence on i and Δ for convenience in the description of the test, the tested hypotheses are:

(H_0): "$\{N(t)\}_{1 \leq t \leq P}$ are independent and identically distributed random variables"

(H_1): "There exists some r such that $(N(1),\ldots,N(r))$ and $(N(r+1),\ldots,N(P))$ have a different distribution".

Let us now describe the test statistic that we use. For each $s,t \in \{1,\ldots,P\}$, we define:

- $A_{s,t} = \mathbb{1}(X(s) > X(t), \delta(s) = 1) - \mathbb{1}(X(s) < X(t), \delta(t) = 1)$, where $\mathbb{1}(E) = 1$, if we are in the event E and 0, otherwise,
- $U_s = \sum_{t=1}^P A_{s,t},\ s = 1,\ldots,P,$
- $S_t = \left(\sum_{s=1}^t U_s\right) / \left(\sum_{s=1}^P U_s^2\right)^{1/2}, t = 1,\ldots,P.$

We shall use

$$W_P = \underset{1 \leq t \leq P}{\text{Max}} |S_t|$$

as a test statistic. Since, under (H_0), see [6],

$$W_P \xrightarrow{D} \|B\|_\infty = \sup_{0<t<1} |B(t)|, \text{ as } P \to \infty,$$

where $\{B(t),\ t \in [0,1]\}$ denotes a Brownian bridge and D the convergence in distribution, we shall take for the change-point detection test the following p-value: $Pval(\text{Max}_{1 \leq t \leq P} |S_t|)$, where for all $b > 0$,

$$Pval(b) = P(\|B\|_\infty > b) = 2 \sum_{j=1}^{\infty} (-1)^{j-1} e^{-2j^2b^2},\ b > 0\ .$$

The last equality is given in [6]. For a given false alarm rate $\alpha \in (0,1)$, we reject (H_0) when $Pval(W_P) < \alpha$. In the rejection case, the change-point instant is given by

$$\hat{t}_P = \underset{1 \leq t \leq P}{\text{Argmax}} |S_t|.$$

3. APPLICATION TO REAL DATA

In this section, we give the results when the *TopRank* algorithm is applied to some real Internet traffic provided by France-Télécom within the framework of the ANR-RNRT OSCAR project. These data correspond to a recording of 67 minutes of ADSL and P2P traffic to which some attacks of type SYN flooding, UDP flooding, PortScan and NetScan have been added. 'UDP flooding' in Figure 1 displays the total number of UDP packets as well as the number of UDP packets received by the destination IP address attacked by UDP flooding. 'SYN flooding' in Figure 1 displays the total number of SYN packets as well as the number of SYN packets received by the destination IP address attacked by SYN flooding. 'PortScan' in Figure 1 displays the total number of TCP packets as well as the number of TCP packets received by the destination IP address attacked by PortScan. Finally, 'NetScan' in Figure 1 displays the total number of packets as well as the number of packets sent by the source IP address generating NetScan.

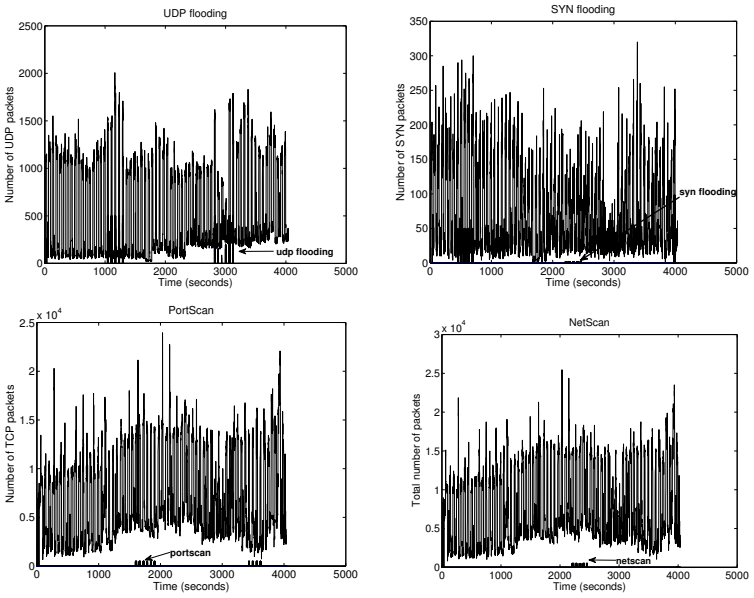

Figure 1: Number of exchanged packets for each type of attack.

From Figure 1, we can see that we are faced with massive data streams and that the attacks are completely hidden and thus difficult to detect.

3.1 Choice of parameters

The previous data have been addressed using the following parameters : $P = 60$, $\Delta = 1$, $M = 10$ and $M' = 1$. With these parameters, a decision concerning the presence of attacks is taken every minute thus entailing an average detection delay of 30 seconds. As for the choice of M', taking $M' = 1$ means that we only analyze the IP addresses i having, at least

once in an observation window, the largest $N_i^{\Delta}(t)$. Note that the reduction stage (Step 1 of *TopRank*) is necessary for the analysis of this dataset. Indeed, Figure 2 displays the number of different destination (left) and source IP addresses (right) each minute of the trace. Thus, applying Step 3 to each IP addresses at each minute would not be feasible on-line.

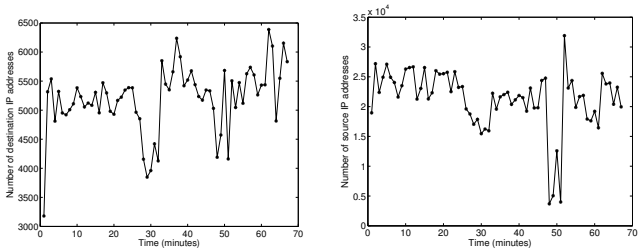

Figure 2: Number of IP addresses every minute.

Figure 3 displays the number of time series which are actually built in Step 2 of *TopRank* after the filtering stage of Step 1 for different values of M' : $M' = 1$, $M' = 5$ and $M' = 10$ when we are interested in NetScan. This means that we are looking for a change in the time series corresponding to the number of destination IP addresses requested by the source IP addresses encountered in each observation window. Step 1 of *TopRank* considerably reduces the number of analyzed time series without removing the attacked IP addresses as we shall see in the following.

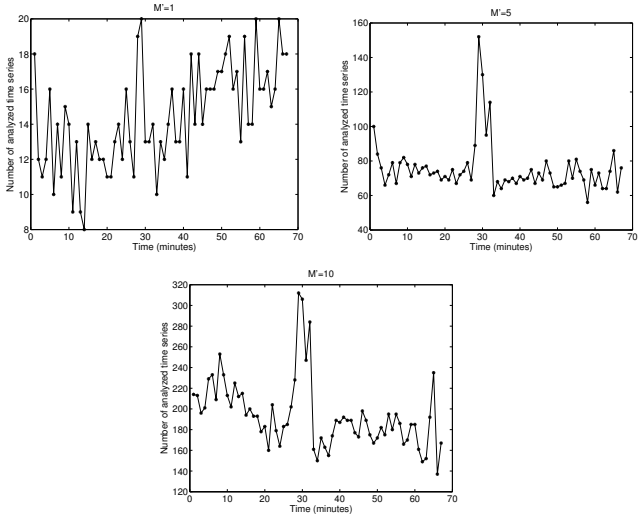

Figure 3: Number of analyzed time series every minute when $M' = 1$, $M' = 5$ and $M' = 10$.

3.2 Performance of the method

3.2.1 Statistical performance

First, note that with the previous choice of parameters the attacked IP addresses have been identified when the upper bound of the p-value α introduced in Step 3 of *TopRank* satisfies $\alpha \geq 10^{-11}$ for the PortScan, $\alpha \geq 10^{-6}$ for the UDP flooding, $\alpha \geq 0.0006$ for the SYN flooding and $\alpha \geq 0.04$ for the NetScan.

Figure 4 displays the censored time series (Step 2 of *TopRank*) of the attacked IP addresses in the case of SYN flooding, UDP flooding and PortScan as well as the censored time series of the source IP address generating the attack in the case of NetScan. These time series are displayed in the first observation window in which the algorithm detects the anomaly of the corresponding type. We also display with a vertical line the instant where the change is detected. The detection time delay is equal to around 1 minute for the SYN flooding, 5 seconds for the UDP flooding, 30 seconds for the PortScan and 20 seconds for the NetScan.

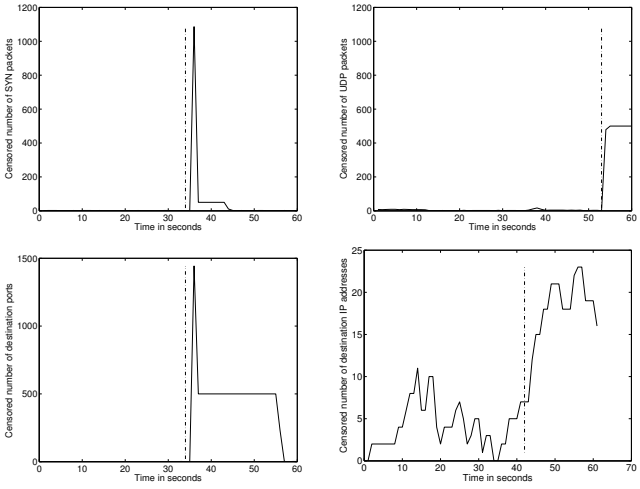

Figure 4: Censored time series of the attacked IP addresses.

We can see from Figure 4 that the rank test for censored data described in Step 3 of *TopRank* can detect several types of changes: sudden increase, slow increase and several types of intensity of changes.

Figure 5 displays ROC curves corresponding to PortScan (left) and TCP/SYN flooding (right). The x-axis of such curves corresponds to the false alarm rate whereas the detection probability is on the y-axis. More precisely, the false alarm rate corresponds to the normalised number of different IP addresses for which an alarm was raised whereas they do not belong to the set of the known attacked IP addresses. The detection probability corresponds to the normalized number of IP addresses which have been detected by the *TopRank* algorithm and which belong to the set of attacked IP addresses. Normalization in the case of the false alarm rate is obtained by dividing the number of false alarms in the sense described above by the number of analyzed IP addresses which are not assumed to be attacked. In the case of PortScan, it is equal to 1922 for the whole trace and to 1867 for SYN flooding. For detection probability, normalization consists in dividing the number of well detected IP addresses by the number of attacked IP addresses.

To compute the false alarm rate and detection probability, we have considered that the attacked IP addresses were only those for which an attack was generated but it is possible that the underlying ADSL traffic contains some attacks. Figure 6 displays the censored time series of some IP addresses which were considered as false alarms in the computation of the ROC curves in the case of the SYN flooding as well as the time instant where a change has been detected (vertical line).

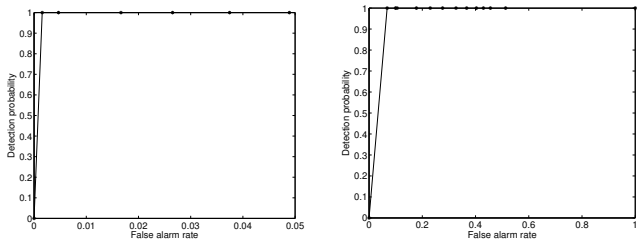

Figure 5: ROC curves for the PortScan (left) and SYN flooding (right).

However, we think that these IP addresses could be considered as being attacked. Thus, the ROC curves are displayed in the worst case for our algorithm.

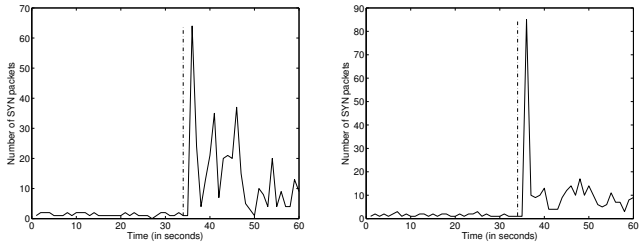

Figure 6: Censored time series for IP addresses considered as false alarm.

3.2.2 Numerical performance

As we have seen, this method seems to give satisfactory results from a statistical point of view. Moreover, with $M = 10$, $M' = 1$, and $P = 60$, applying the *TopRank* algorithm takes only 1 minute and 30 seconds to process the whole trace of 67 minutes, which means looking for SYN flooding, UDP flooding, PortScan and NetScan, with a computer having the following configuration: RAM 1 GB, CPU 3 GHz. This makes the on-line implementation of *TopRank* very realistic even for more intense traffic data.

4. CONCLUSION

In this paper, we propose an intrusion detection method for identifying DoS and DDoS attacks in Internet traffic: the *TopRank* algorithm based on a record filtering technique followed by a nonparametric rank test. The *TopRank* algorithm turns out to be a very efficient technique to detect several types of flooding attacks both from a statistical and numerical point of view. More precisely, the main attractive features of the *TopRank* algorithm are twofold. First, it is able to adapt to various types of traffic on account of the nonparametric property of the test stage. Secondly, its computational simplicity and efficiency make its on-line implementation feasible. This is already the case: our algorithm is integrated in an experimental network developed by France-Télécom R&D and the other partners involved in the ANR-RNRT OSCAR project and the results obtained with our method seems to be very promising.

REFERENCES

[1] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes:Theory and Applications*. Prentice-Hall, 1993.

[2] B. E. Brodsky and B. S. Darkhovsky. *Nonparametric Methods in Change-Point Problems*. Kluwer Academic Publisher, 1993.

[3] CISCO. http://www.cisco.com.

[4] M. Csörgo and L. Horvath. *Limit theorems in change-point analysis*. Wiley, New-York, 1997.

[5] E. Gehan. A generalized Wilcoxon test for comparing arbitrarily single censored samples. *Biometrika*, 52:203–223, 1965.

[6] E. Gombay and S. Liu. A nonparametric test for change in randomly censored data. *The Canadian Journal of Statistics*, 28(1):113–121, 2000.

[7] B. Krishnamurthy, S. Subhabrata, and Y. Zhang. Sketch-based change detection: methods, evaluation and applications. IMC'03, 2003.

[8] A. Lakhina, M. Crovella, and C. Diot. Diagnosing network-wide traffic anomalies. Proc. of SIGCOMM'04, 2004.

[9] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina. Detection and identification of network anomalies using sketch subspaces. pages 147–152. Proceedings of the 6th ACM SIGCOMM on Internet measurement, 2006.

[10] N. Mantel. Ranking procedures for arbitrarily restricted observations. *Biometrics*, 23:65–78, 1967.

[11] E. S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[12] V. Paxson. Bro: A system for detecting network intruders in real-time. *Computer Network*, 31(23–24):2435–2463, 1999.

[13] M. Roesch. Snort: Lightweight intrusion detection for networks. pages 229–238. 13th Syst. Admin. Conf. (LISA), 1999.

[14] A. Siris and F. Papagalou. Application of anomaly detection algorithms for detecting SYN flooding attacks. ICON 2004 - 12th IEEE International Conference on Network, 2004.

[15] A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim. Detection of intrusion in information systems by sequential change-point methods. *Statistical Methodology*, 3(3):252–340, 2006.

[16] A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE Transactions on Signal Processing*, 54(9), 2006.

[17] H. Wang, D. Zhang, and G. Shin. Detecting SYN flooding attacks. Proc. of IEEE INFOCOM'02, 2002.