A DoS Detection Method Based on Composition Self-Similarity

Zhu Jian-Qi[1,2], Fu Feng[1], Chong-kwon Kim[2], Yin Ke-xin[3] and Liu Yan-Heng[1]
[1] College of Computer Science and Technology, Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, 2699 Qianjin Street, Changchun, China
[2] School of Computer Science and Engineering, Seoul National University
[3] College of Software, Changchun University of Technology, 2055 Yanan Street, Changchun, China
[e-mail: zhujq@jlu.edu.cn, fgbest@sina.com, lyh_lb_lk@yahoo.com.cn, ckim@snu.ac.kr, yinkexin@126.com]
*Corresponding author: Chong-kwon Kim and Liu Yan-Heng

Received February 3, 2012; revised April 16, 2012; accepted May 7, 2012;
published May 25, 2012

Abstract

Based on the theory of local-world network, the composition self-similarity (CSS) of network traffic is presented for the first time in this paper for the study of DoS detection. We propose the concept of composition distribution graph and design the relative operations. The $(R/S)^d$ algorithm is designed for calculating the *Hurst* parameter. Based on composition distribution graph and Kullback Leibler (KL) divergence, we propose the composition self-similarity anomaly detection (CSSD) method for the detection of DoS attacks. We evaluate the effectiveness of the proposed method. Compared to other entropy based anomaly detection methods, our method is more accurate and with higher sensitivity in the detection of DoS attacks.

***Keywords*:** DoS detection, composition self-similarity, composition distribution graph, Kullback-Leibler divergence

This work is supported by the National Natural Science Foundation of China under Grant No.60973136, 61073164; Next-Gen Internet Service Trial Commercialization and Equipment Industrialization Projects of National Development and Reform Commission (NDRC) under Grant No. CNGI-09-01-11; Innovation Project of Jilin University under Grant No.450060445169; International Cooperation and Exchange Project of State Commission of Science Technology of China under Grant No.2008DFA12140; Erasmus Mundus External Cooperation Window under Grant No.155776-EM-1-2009-1-IT-ERAMUNDUS-ECW-L12; Youth Foundation of Jilin Province Science and Technology Development under Grant No.201101033.

http://dx.doi.org/10.3837/tiis.2012.05.012

1. Introduction

Because of its becoming the major universal communication infrastructure, Internet is also subject to attacks in growing numbers and varieties, notably, Denial of Service (DoS) attack has become one of the most serious threats to the Internet [1]. It doesn't necessarily damage data directly or permanently, but it intentionally compromises the availability of the resources. *Misuse detection* [2][3] uses the "signatures" of known attacks to identify a matched activity as an attack instance and strictly limited to the latest known attacks. *Anomaly detection* [4] can be effective against new attacks. However, due to the lack of theoretical understandings and useful tools for characterizing the audit data, most *anomaly detection* models are built based solely on "expert" knowledge, which is often imprecise and incomplete given the complexities of today's network environments. *Volume-based detection* schemes have been successful in isolating large traffic changes, but a large class of anomalies do not cause detectable disruptions in traffic volume (e.g., scans or small DoS attacks may have a minor effect on the traffic volume of a backbone link). Moreover, the similarity between DoS traffic and transient bursts of normal traffic makes it difficult to detect DoS attacks accurately.

Leland et al [5] first convincingly demonstrated that actual network traffic is statistically *self-similar* in nature that none of the commonly used traffic models (e.g., pure Poisson or Poisson-related models such as Poisson-batch or Markov-Modulated Poisson processes) is able to capture this *fractal-like* behavior. Following up this prominent discovery, the works by [6][7][8] brought self-similar process to the applications of DoS detection. However, as described in [5], this self-similarity is confined to the traffic *value*. In other words, the statistically self-similar models in these papers only focus on the changes of traffic *value* not considering the impact of distributional aspects of packet (called packet composition). Intuitively, DoS attacks are purposely created by humans they must affect the natural "structure and randomness" of packet under normal conditions [9]. Jiangtao Shi [10] controlled the number of packets with statistically *self-similar* model, however, the generation of IP address and Port is based on probabilities. Moreover, [6][7][8] on DoS detection via self-similarity estimation are based on the assumption that the DoS traffic is not self-similar. However, from the review of existing literature and the analysis of software tools used to mount DoS attack, it can be inferred that these tools are capable of generating a self-similar flow of packets. One question then becomes: if both legitimate and malicious traffic display statistically self-similar characteristics, is detection of DoS still possible? An important challenge therefore is to determine how best to extract *understanding* about the presence and nature of DoS attack from the potentially overwhelming mass of network-wide traffic.

Our work begins with the observation that most traffic anomalies share a common characteristic: they induce a change in distributional aspects of packet header fields. For

example, a DoS attack, regardless of its volume, will cause the distribution of traffic by destination addresses to be concentrated on the victim address. Treating anomalies as events that disturb the distribution of packet *compositions* has more advantages against previous methods (e.g., *volume-based* method). First, such anomalies as scans or small DoS attacks can be better detected by systematically mining for distributional changes instead of volume changes. Second, unusual distributions reveal valuable information about the structure of anomalies. Current research in the traffic composition mainly relies on the theory of entropy [9][11][12][13] by which network anomalies can be detected. However, the flaw of entropy methodology is that it can only reflect the overall trend of traffic and not sensitive to the dynamics of traffic specific composition. Moreover, the entropy of traffic composition is unstable. In the following sections, we will further illustrate the advantages of our presented method compared with the entropy based method, for example, the presented method has higher detecting rate, lower false alarm and higher sensitivity to DoS attacks.

This paper focuses on examining distributional features of packet compositions that produces effective diagnostic power in anomaly detection. The concept of traffic *composition self-similarity* (CSS) is proposed for the first time based on the local-world network, and we validate the existence of CSS feature in network-wide traffic. Then we design a $(R/S)^d$ algorithm for computing *Hurst* parameter (the degree of self-similarity typically is defined via the *Hurst* parameter). In order to detect DoS attacks, we introduce the concept of *composition distribution graph* and define the relative operations. With the composition distribution graph and KL divergence, we design the *composition self-similarity* anomaly detection (CSSD) method for the detection of DoS attacks. We illustrate and evaluate the proposed method by using some well-known attacks and compare the results with other popular entropy based methods. The results show that our method is more accurate and effective.

The remainder of this paper is structured as follows. Section 2 defines the concepts of *composition self-similarity* of traffic and composition distribution graph, and designs a $(R/S)^d$ algorithm for the estimation of *Hurst* parameter. Section 3 describes the detection principle of CSSD method and gives the algorithm. Section 4 presents the experiment and analysis. Section 5 concludes this paper.

2 Composition Self-Similarity

2.1 KL Divergence

The Kullback-Leibler (KL) divergence [14] measures the distance between two density distributions. This divergence is also known as information divergence and relative entropy. Consider two discrete probability distributions, p_i and q_i, with $\sum p_i = \sum q_i = 1$. The KL divergence between these two distributions is defined as:

$$D(P \| Q) = \sum_i p_i \log(p_i / q_i) \quad (1)$$

When $p_i = q_i$ (i=1,2,…,n), $D(P||Q)$=0. The smaller divergence, the more "similar" they are.

2.2 Composition Self-Similarity of Traffic

The statistical *self-similarity* feature in [5] shows that the traffic distribution is similar irrespective of the time scale of observation, but it only cares about the *value* changes of traffic. On the contrary, this paper will study the dynamics of traffic composition (i.e., source and destination address and ports) for the detection of anomalies, the intuition of which is that many kinds of traffic anomalies cause changes in the distribution of addresses or ports observed in traffic. The traffic composition here only studies the packet header that includes source IP, destination IP, source port and destination port. [10] points that whatever abnormality appears that will definitely affect the distribution of destination IP. So, taking into account the computational complexity and real-time requirements, we will only focus on the CSS feature of destination IP (denoted *dstIP*).

The set $S=\{ip_1,ip_2,...,ip_L\}$ denotes the *dstIPs* of all packets. First, S is partitioned and it produces a sequence X. In order to reduce the impact of sudden network traffic, the partition is based on the number of packets instead of the time series. Take C consecutive *dstIPs* as a data point (denoted X_i) and thus S is converted into the sequence $X=\{X_1,X_2,...,X_N\}$, where $X_i=\{ip_{(i-1)c+1},...,ip_{i*c}\}$ and $N=ceil(L/C)$. In the following, we will introduce the concept of composition distribution graph.

Definition 1 Composition distribution graph G. G is a composition distribution graph that illustrates the IP's appearance rate in one or more data points. G_i is the composition distribution graph of X_i, $G_i=\{(ip^i_1,p^i_1),...,(ip^i_k,p^i_k)\}$ means that there are k *dstIPs* in X_i, p^i_j denotes the appearance rate of ip^i_j.

Fig. 1 is the graphical representations of G_5 and G_6 in polar coordinate based on DARPA99 (X is the outside.dump of March 1 in dataset, G_5 and G_6 are computed through X_5 and X_6 when C=2000). In this figure, the radius coordinate consists of the appearance rates of *dstIPs* and the angle coordinate consists of permutation of all the *dstIPs*. The feature of CSS is to study the relationships of composition distribution graphs of all data points and the relationships between local and global composition distribution graph. CSS means that the network traffic displays structural similarity in a very wide time scale from the point of traffic composition distribution. The real networks are mostly complex systems which contain lots of members and connections. These members are abstracted to nodes and connections are abstracted to edges. In [15], a local-world evolving network model is proposed by Li and Chen. The study shows that in the real network, a node can only connect to special group of nodes rather than any node in the whole network. As in a human community, everyone literally lives in a local world: everyone has his local friendships and personal collective information and judgment. The local-world network model will enable us to better understand and describe more real-life complex networks. In a certain period of time, the users and accessing services in a local-world community is stable that makes the traffic composition stable. That is, the network traffic exhibits the CSS feature.

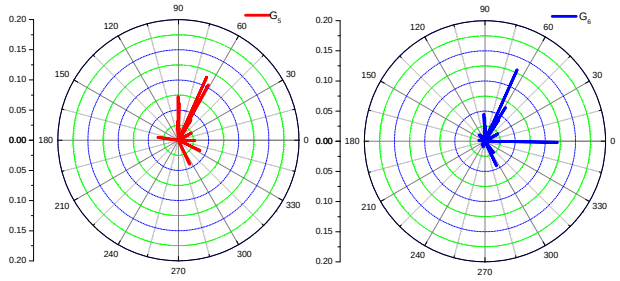

Fig. 1. Examples of composition distribution graph (G_5 and G_6)

Definition 2 The CSS feature of network traffic is structural similarity in the composition ratio across all time and spatial scales. It can be described by the composition skeleton graph (denoted G_M).

Definition 3 Composition skeleton graph G_M. The feature of CSS means that traffic exhibits a stable structure over different scales, in this paper this stable structure is called composition skeleton and denoted as G_M.

With this in mind, the feature of CSS can be viewed that the composition distribution graphs of traffic are similar among each other over different scales and they are all similar with G_M. *Hurst* is an important index to measure the similarity of series X. When $0.5<Hurst<1$ it means that X is self-similar. As *Hurst* increases, the degree of self-similarity is increasing. For $Hurst=0.5$ and $Hurst>1$, there is almost no self-similarity.

2.3 (R/S)d Algorithm for Hurst Estimation

The R/S method in [16] is a traditional method for *Hurst* estimation of one-dimensional data that only can analyze the value changes of traffic. In order to analyze the similarity of the traffic spatial composition distribution, in this paper, we present an extensional method called $(R/S)^d$. The reason that $(R/S)^d$ is used to analyze the feature of similarity is that $(R/S)^d$ can calculate the multi-dimensional data.

The operating rules of composition distribution graph are defined as follows: for each (ip_i,p_i), all operations are carried on p_i , ip_i is the operation object.

Definition 4 Addition of composition distribution graphs $G_{i,j}=G_i+G_j$. For each $(ip^{i,j},p^{i,j})$ in $G_{i,j}$, $p^{i,j}=p^i+p^j$, p^i and p^j are the appearance ratios of *dstIP* in G_i and G_j, respectively.

Definition 5 Union of composition distribution graphs $GE_{i,j}=G_i \cup G_j$. The union operation will combine X_i and X_j to form a new IP address series $X_{i,j}$. All the IP addresses and their appearance ratios in $X_{i,j}$ are denoted by $GE_{i,j}$.

Corollary 5.1 $GE_{i,j}=(1/2)G_{i,j}$. $(1/2)G_{i,j}$ means that each p in $G_{i,j}$ will be multiplied by 0.5, and $GE_{i,j}$ can be viewed as the mean of G_i and G_j.

Proof From Definition 4, any IP that appears in $GE_{i,j}$ must be in $G_{i,j}$. For any (ip_k,p_k) in $GE_{i,j}$, $p_k=c_k/2C$, c_k is the occurrence number of ip_k in $X_{i,j}$. The data pair $(ip_k,p^{i,j})$ in $G_{i,j}$ is

corresponding to ip_k, where $p^{i,j}=p^i+p^j= c_i/C+ c_j/C$, c_i and c_j are the occurrence numbers of ip_k in X_i and X_j. So, $c_k= c_i+ c_j$, $p^{i,j}= c_k/C= 2p_k$, and $GE_{i,j}=(1/2)G_{i,j}$.

Corollary 5.2 $G_{i,j,k}=G_i+G_j+G_k$ and $GE_{i,j,k}=(1/3)G_{i,j,k}$.

Proof $G_i+G_j+G_k = G_{i,j}+G_k = G_{i,j,k}$. $GE_{i,j,k}=(1/3)G_{i,j,k}$, the proof process is similar to Corollary 5.1.

Definition 6 Subtraction of composition distribution graphs G_i-$G_j=D(G_i||G_j)$. The computation process refers to formula (1), the result of subtraction is the difference of G_i and G_j.

The $(R/S)^d$ algorithm is described as: first, according to C, S produces the sequence $X=\{X_1,X_2,...,X_N\}$, where X_i can be represented as G_i; second, divide X into M sub-sequences $Y_m=\{X_{(m-1)d+1},X_{(m-1)d+2},...,X_{md}\}$ (m=1,2,3,...M), where $M=N/d$. Different $(R/S)_d$ can be computed according to different d (d=1,2,3,...M/2); last, fitting the R/S line with all the $(R/S)_d$ values in order to get the *Hurst* parameter.

Compute GE_m (the mean of sub-sequence Y_m) as formula (2).

$$GE_m = \frac{1}{d}\sum_{i=(m-1)d+1}^{md} G_i \tag{2}$$

GE_m is the mean composition distribution graph of all IP addresses in Y_m; Compute the standard deviation of Y_m as (3), which is denoted as S_m.

$$S_m = \frac{1}{d}\sum_{i=(m-1)d+1}^{md} (G_i - GE_m)^2 \tag{3}$$

According to Definition 3, (3) can also be expressed,

$$S_m = \frac{1}{d}\sum_{i=(m-1)d+1}^{md} (\mathrm{D}(G_i \parallel GE_m))^2 \tag{4}$$

Compute the cumulative deviation $S_{i,m}$ as (5).

$$S_{i,m} = \sum_{j=1}^{i} G_j^m - iGE_m \tag{5}$$

Where i and m denote the i-th element in Y_m (i=1,2,...,d). The complete representation of $G^m{}_j$ in X is $G_{(m-1)d+1+j}$, which is the composition distribution graph of $X_{(m-1)d+1+j}$. The range R_m is computed based on the cumulative deviation.

$$R_m = \max\{S_{1,m},...,S_{d,m}\} - \min\{S_{1,m},...,S_{d,m}\} \tag{6}$$

Compute $(R/S)_d$, which is the average value of R_m/S_m of all subsequences,

$$\left(\frac{R}{S}\right)_{\mathrm{d}} = \frac{1}{M}\sum_{m=1}^{M}\frac{R_m}{S_m} \tag{7}$$

Different d corresponds to different $(R/S)_d$, from a statistical point of view, the relation between them can be represented by $(\mathrm{R/S})_{\mathrm{d}} \sim cd^H$, taking the log of both sides,

$$\log\left(\frac{R}{S}\right)_{\mathrm{d}} = \log c + H\log d \tag{8}$$

Where, log$c$ is a constant. Depict all the points (log$d$,log$(R/S)_d$) in the logarithmic coordinates and get the slope H by linear fitting, which is the *Hurst* parameter. As shown in **Fig. 2**.

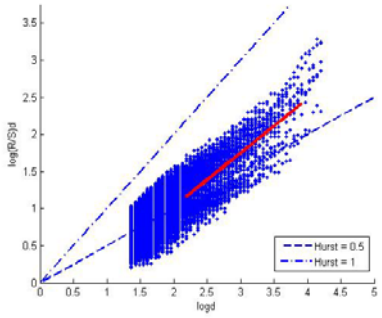

Fig. 2. *Hurst* parameter in $(R/S)^d$ algorithm

In **Fig.2**, each node represents a $(R/S)_d$ based on different d. The two dotted lines represent *Hurst*=0.5 and *Hurst*=1, respectively. The series X holds the CSS feature when the calculated slop is within the limits of 0.5 and 1.

3. Detection Principle and Algorithm

3.1 Detection Principle

From the composition distribution point of view, the traffic exhibits a stable structure over different scales that can be represented with G_M. In this paper, we assume that if the network is normal at t_{i-1} time and no great changes happen at t_i time, we consider that the network is also normal at t_i time. To facilitate presentation, we define $CM_i = D(G_i||G_M)$ that represents the difference between G_i (at t_i time)and G_M. CM_i is also represented by CM with no ambiguity. In the same way, $CP_i = D(G_i||G_{i-1})$ that is used to represent the difference between the adjacent composition distribution graphs. Also, CP_i is denoted as CP used to identify if mutations occur between the adjacent time intervals.

In the following, we will demonstrate the application of KL divergence and G_M to detect DoS attacks. The schematic diagram of composition self-similarity detection (CSSD) in this paper is shown in **Fig. 3**, which is composed of the skeleton distribution graph extraction module and the KL divergence calculation module. The former is an off-line analysis module that extracts G_M by analyzing the traffic, and the latter is an on-line real time analysis module for calculating CP and CM. There are basically four steps involved from the observation of traffic stream to raise an alarm. The first is to gather information from streaming traffic. The traffic monitoring module observes packet's headers and collects information in order to calculate G_M. The accuracy of G_M is greatly affected by abnormal traffic. In this paper, we first calculate multiple G_{Mi} (i=1,2,.....,n) and compute its mean G_{EM}, from which select $MIN(G_{Mi}\text{-}G_{EM})$ as the traffic composition skeleton graph, that is, $G_M = MIN(G_{Mi}\text{-}G_{EM})$. The second step is to quantify variations observed in the composition distribution. The third is to

set a threshold that can clearly differentiate DoS attacks from normal behaviors. The fourth step is to gather all the information and identify the attacking source.

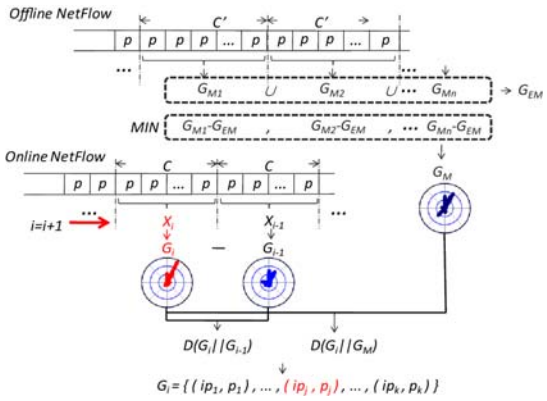

Fig. 3. CSSD schematic diagram

The detection result of CSSD is shown in **Fig. 4**, in which *x*-axis and *y*-axis represents *CP* and *CM*, respectively**.** The thresholds δ_M and δ_N divide the coordinate into Z_1, Z_2, Z_3 and Z_4 sub-regions. The points in Z_1 are normal because the values of *CP* and *CM* are in a certain range. The points in Z_2 have mutations, but it is also normal because *CM* is still in a certain range. Z_3 represents the abnormal, in which not only mutations occur but also *CM* is greater. When the anomaly occurs, the abnormal points would fall into Z_3. With the anomaly continuing, the points would fall into Z_4. For example, when the network is subjected to DoS attacks, the current G_i will have so big differences with both G_{i-1} and G_M that the points would fall into Z_3. As DoS attacks continue, *CP* is small, but *CM* is still large so that all the points would fall into Z_4.

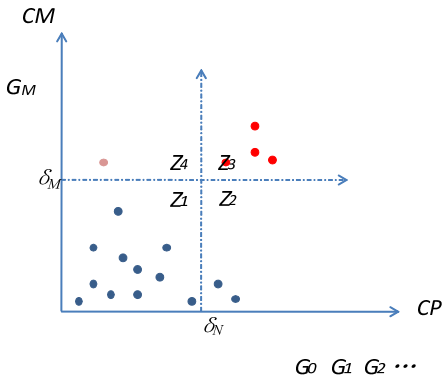

Fig. 4. The detection principle of CSSD

3.2 The Detection Algorithm

The core of CSSD method is the on-line anomaly detection. The online study of traffic composition distribution is a difficult task mainly due to vast amount of data and limited

computational resources. To overcome these problems, our method introduces two techniques: (1) The network traffic compositions are aggregated at a very coarse level. In this paper, we only take into account the destination IP. (2) The variations of traffic composition distribution are measured using simple and efficient KL divergence algorithm.

The jobs of KL divergence calculation module are: (1) calculates *CP* and *CM*; (2) backtraces IPs of the points in Z_3 in order to find the main IP that leads to the changes of traffic composition distribution. This module is a stepping cycle calculation process. In each cycle, it first completes calculating the current G_i with G_{i-1}, and then calculating *CP* and *CM*. If it detects the abnormal, it will enter into the IP backtracing process. G_i will be the input in the next cycle, and the algorithm is described as follows:

Step1. Initialize the parameters. Set *C* and *i,* and obtain the value of G_M;

Step2. Set the current *C dstIPs* as a data point, compute G_i;

Step3. Compute *CP* and *CM*;

Step4. If $CP > \delta_N$ and $CM > \delta_M$, mark the current data point as attack. Otherwise, go to Step6;

Step5.Backtrace IP and compute $IP_{G_{i-1}}$ and IP_{G_m} ;

Step6. $i=i+1$, go back to Step2;

G_i and G_{i-1} are needed in the computation of *CP*. However, G_{i-1} is obtained in the last cycle so that it reduces the complexity. The IP backtracing in Step5 is described as follows, taking the computation of $IP_{G_{i-1}}$ as an example:

Step1. Assume G_i contains k IP, then $G_i=\{(ip^i_1,p^i_1),...,(ip^i_k,p^i_k)\}(j=1)$;

Step2. If $j \le k$, set p^i_j in G_i as the ratio of ip^i_j in G_{i-1}, and we get a new G_i^{*j};

Step3. Compute $D(G_i^{*j} \| G_{i-1})$;

Step4. $j=j+1$ and go to Step2;

Step5. $D=\max\{D_1,D_2,...,D_j\}$, the IP that makes *D* reach the maximum is $IP_{G_{i-1}}$. The algorithm is over.

The IP that leads to the greatest changes of the composition distribution graph will cause anomaly. If the values of multiple IPs are similar and do not have a clear maximum, it means that multiple hosts appear abnormal.

4. Experiment Analysis

We extracted five weeks (25 days) of data from DARPA99 datasets, in which the first three weeks of data are training data and the other two weeks are testing data. The first and the third weeks contained no attacks, and the other three weeks are abnormal traffic. We compute every day's self-similarity parameter *Hurst* that is plotted as a dot in **Fig. 5**. *desHurst* and *souHurst* denote the *Hursts* of *dstIP* and *srcIP*, respectively**.** It can be seen that

the *Hursts* of source and destination are both greater than 0.5 in the attack-free network environment and these two *Hursts* are close. The result shows that the CSS feature exists in the local-world network traffic.

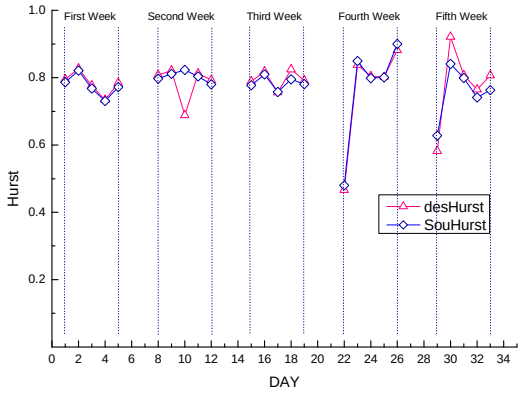

Fig. 5. *Hurst* curves of CSS

The *Hurst* values in the first and the third week are stable; however, *Hurst* changes a lot in the presence of attacks**.** The three lowest points in *desHurst* curve are $Hurst_{3.10}$=0.6881, $Hurst_{3.29}$=0.4668 and $Hurst_{4.5}$=0.5821. The two lowest points in *souHurst* curve are $Hurst_{3.29}$=0.4802 and $Hurst_{4.5}$=0.6277. It can be seen that the anomalies in the network will affect *Hurst*, but not all anomalies can cause the change of *Hurst* because *Hurst* describes the overall property of traffic. For example, if attack does not cause the entire anomaly in network or the time period of anomaly is very short, *Hurst* will not be greatly affected.

Studying the traffic of March 1 from outside.dump, different *Hurst* is computed according to different *C* (from 2000 to 20000) as shown in **Fig. 6**. **Fig. 7** and **Fig. 8** are the composition distribution graphs when *C*=2000 and *C*=20000.

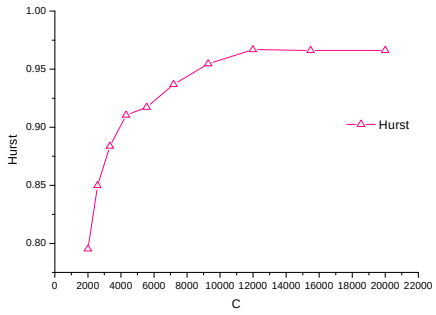

Fig. 6. *Hurst* curve corresponds to *C*

Fig. 6 shows that *Hurst* increases with *C* increasing and gradually stabilizes that makes the overall CSS feature increase and finally tends to a certain value H_F. The relation between *C* and H_F are relative to the size of local-world network (number of hosts). The CSS feature is

stronger when C=2000, similarly, in a larger network, the reflection of CSS feature depends on greater C.

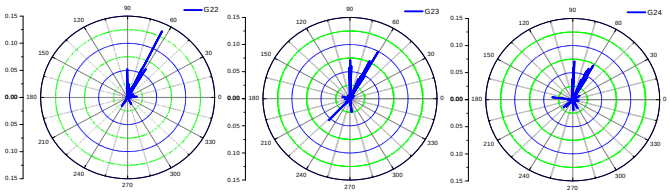

Fig. 7. The composition distribution graph (C=2000)

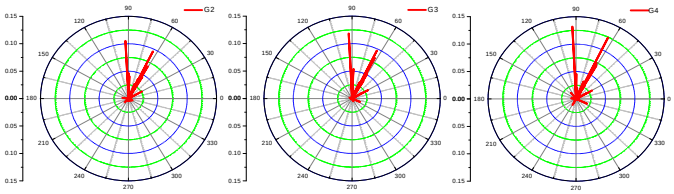

Fig. 8. The composition distribution graph (C=20000)

In **Fig. 7**, G_{22}, G_{23} and G_{24} are similar. In **Fig. 8**, the similarity of G'_2, G'_3 and G'_4 is further enhanced because C and *Hurst* are greater. From **Fig. 6**, **Fig. 7**, and **Fig. 8**, it shows that the greater C the stronger CSS feature of traffic. However, the detecting effect does not depend on greater C. Because greater C will definitely increase the time span of G, and with longer computation, that will lead to a larger detection delay. Moreover, greater C could lead to an increment in false negative. On the contrary, smaller C could produce more false positive (because G could not accurately reflect the feature of traffic composition distribution). In this paper, we choose a greater C when we extract G_M. When computing CP and CM, in order to achieve a better detection effect, we appropriately adjust C, δ_N and δ_M according to the real network environment.

In the skeleton distribution graph extraction module, G_M can be obtained from the historical data**.** As shown in **Fig. 9**, G_M in the first and the second week are basically same, which does not mean that any G_M in the second week can be used as the network composition skeleton graph. The accuracy of G_M is greatly affected if more abnormal traffic is contained. As described in 3.1, $G_M = MIN(G_{Mi}\text{-}G_{EM})$. The CSSD method is noise immunity because it does not require completely pure data in the training process. In the real network, G_M can be updated periodically, for example, we can choose the appropriate G_M in different times when the traffic composition changes with time periodically.

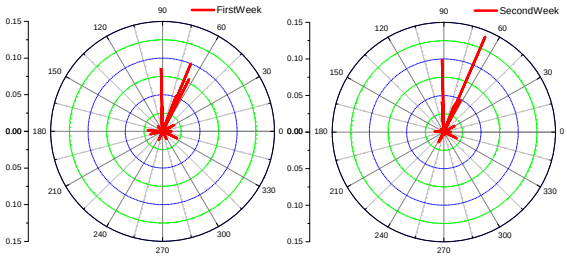

Fig. 9. Comparison of choosing G_M

In this experiment, we choose the traffic of April 5 that contains more DoS attacks, and compare the CSSD with the method in [12]. As shown in **Fig. 10**, the four curves represents *CM*, *CP*, IP entropies of destination and source from top to bottom, respectively.

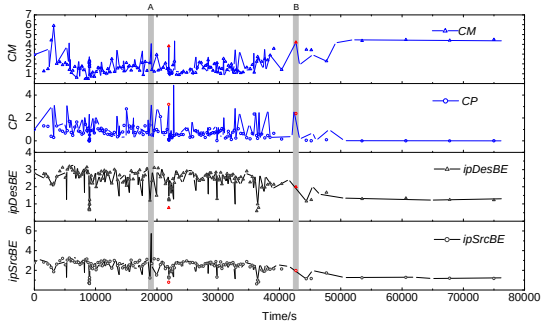

Fig. 10. Comparison of IP entropy and CSS difference

A represents the Smurf attack broken out at 19091 second, which makes the innocent hosts receive a great deal of ICMP messages from others in a short period of time. From the entropy point of view, Smurf attack can cause the *dstIP* concentrated so that the entropy decreases (value of *ipDesE*), however, the *srcIP* will be divergent that leads to the entropy increase (value of *ipSrcE*). From the CSS point of view, the distribution of traffic composition will change a lot at this moment. **Table 1** is the comparisons between T_p and T_c (T_p represents the previous moment and T_c represents the moment when attack starts).

Table 1. Comparison between CSS difference and entropy

	ipSrcE	***ipDesE***	***CP***	***CM***
T_p	3.1623	1.4114	1.2840	1.7793
T_c	5.9200	0.3374	3.2749	4.1916

Based on the backtracing algorithm, the IP 172.16.114.50 that makes the biggest changes in *CM* is the victim host**.** *B* represents the Udpstorm attack happened in 43225 second. UdpStorm is a bandwidth consumption attack that makes the innocent host sends data from each other continuously by forging the UDP packets. The entropy has no significant changes because both source and destination IP have no obvious concentration or divergence

phenomenon when the attack happens. From the CSS point of view, though IP has no great changes, the proportion of each IP changes a lot when compared with the previous or normal moment. **Table 2** lists the IP changes and the parameters.

Table 2. IP changes and comparisons among the parameters

	IP number	***Same IP***	***ipSrcE***	***ipDesE***	***CP***	***CM***
T_p	25	22	1.8545	1.8674	0.3713	2.3728
T_c	27		2.1190	2.1328	2.6592	3.3539

The essential difference between CSSD and entropy method is that the CSSD method not only considers the macro distribution but also cares about the changes of each composition. The detection result of outside.dump on April 5 is shown in **Fig.11**.

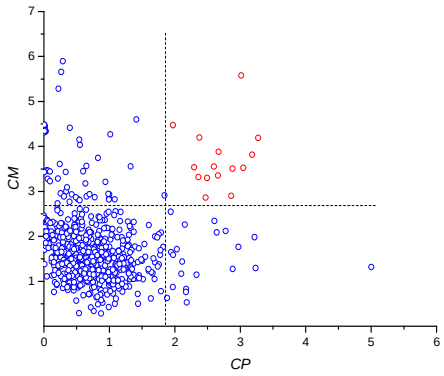

Fig. 11. CSS based DoS detection result

In **Fig. 11**, the detection effect is the best when δ_N=1.9 and δ_M=2.7 so that they can be used in the DoS detection. Repeat each one of 17 attacks 10 times to form a DoS dataset, and insert them into the first and the third week normal traffic randomly. Then verify the CSSD and entropy methods, respectively, the results are shown in **Table 3**.

Table 3. Comparison of CSSD and entropy based methods

	Attack number	**Detection rate**	**False negative**	**False alarm**
CSS	170	95.8%	4.2%	1.7%
Entropy	170	70.6%	29.4%	4.2%

The results show that the CSSD method has a higher detection rate and lower false alarm. In order to compare their detecting sensitivities, we inject the Smurf traffic into the normal traffic successively with an increment of 10%. Compare the changes of *CM*, *CP* and *ipDesBE* as shown in **Fig. 12**. In the detecting process, δ_N=1.9, δ_M=2.7 and δ_E=1.3. It can be seen that the CSSD method can detect the attack when the abnormal traffic is less than 40%

of overall traffic. However, the entropy method requires more than 60% of traffic. That is, the CSSD method has a higher sensitivity.

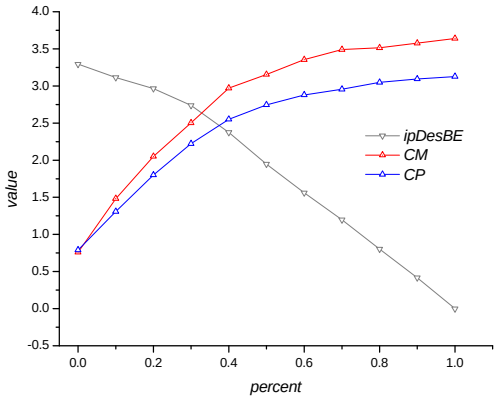

Fig. 12. CSSD based DoS detection result

5. Conclusion

This paper studies the characteristics of traffic composition in the local-world network environment, and presents an effective method for detecting DoS attacks. The main innovations are: (1) first presents the CSS feature of network traffic; (2) defines the composition distribution graph G and the relative operations, designs $(R/S)^d$ algorithm for calculating parameter *Hurst*; (3) presents the CSSD based detection method for detecting DoS attacks.

The experiment shows that the CSSD based method not only has a higher detection rate, but also has a lower false alarm and more sensitivity compared with the entropy based method. This paper only considers the DoS attacks. The next step, we will study the effects of other attacks on the CSS feature, and further on which construct an anomaly model.

References

[1] J Mirkovic and P Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Computer Communications Review*, vol.34, no.2, pp.39-53, Apr.2004. Article (CrossRef Link)

[2] S.Kumar and E.H.Spafford, "A software architecture to support misuse intrusion detection," in *Proc. of 18th National Information Security Conference* , pp.194-204, Oct.1995.

[3] K.Ilgun, R.A.Kemmerer and P.A. Porras, "State transition analysis: a rule-based intrusion detection approach," *IEEE transactions on software engineering*, vol.21, no.3, pp.181-199, Mar.1995. Article (CrossRef Link)

[4] T.Lunt, A.Tamaru, F.Gilham, R.Jagannathan, P.Neumann, H.Javitz, A.Valdes and T.Garvey, "A real-time intrusion detection expert system (IDES)—final technical report," *Computer science library*, SRI International, Menlo Park, California, Feb.1992.

[5] Leland et al., "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions of Networking*, vol.2, no.1, pp.1-15, Feb.1994. Article (CrossRef Link)

[6] W.H. Allen and G.A. Marin, "The loss technique for detecting new Denial of Service attacks," in *Proc. of Southeast Conference,* pp.302-309, Mar.2004.

[7] Y. Xiang, Y. Lin, W.L. Lei and S.J. Huang, "Detecting DDoS attack based on network self-similarity," in *Proc. of IEEE Communications*, vol.151, no.3, pp.292-295, Jun.2004. Article (CrossRef Link)

[8] Ming Li, "Change trend of averaged Hurst parameter of traffic under DDoS flood attacks," *Computers & Security*, vol.25, no.3, pp.213-220, May.2006. Article (CrossRef Link)

[9] Lawniczak AT, Wu H and Di Stefan BN, "Detection of anomalous packet traffic via entropy," in *Proc. of 22nd IEEE Canadian Conference on Electrical and Computer Engineering*, pp.137-141, May.2009.

[10] Lakhina A, Crovella M and Diot C, "Mining anomalies using traffic feature distributions," *Computer Communication Review*, vol.35, no.4, pp.217-228, Oct.2005. Article (CrossRef Link)

[11] E. Earl Eiland and Lorie M. Liebrock, "An application of information theory to intrusion detection," in *Proc. of 4th IEEE International Workshop on Information Assurance*, pp.119-134, Apr. 2006.

[12] Nychis G, Sekar V and Andersen DG, "An empirical evaluation of entropy-based traffic anomaly detection," in *Proc. of 8th ACM SIGCOMM Internet Measurement Conference*, pp.151-156, 2008. Article (CrossRef Link)

[13] Rahmani H, Sahli N and Kammoun F, "Joint entropy analysis model for DDoS attack Detection," in *Proc. of 5th International Conference on Information Assurance and Security*, pp.267-271, Aug.2009.

[14] Thomas M and Joy A, *Elements of Information Theory*, John Wiley & Sons Inc., New York, 2006.

[15] Xiang Li and G. Chen, "A local-world evolving network model," *Physical A*, vol.328, no.1-2, pp.274-286, Oct.2003. Article (CrossRef Link)

[16] Park C, Hernandez-Campos F and Le L, et al, "Long-range dependence analysis of Internet traffic," *Journal of Applied Statistics*, vol.38, no.7, pp.1407-1433, 2011. Article (CrossRef Link)

Zhu Jian-qi received the B.S. degree in Computer Software and M.S. degree in Computer Application Technology in Jilin University in China, in 1999 and 2004, respectively. He also received the Ph.D. degree in Computer Application Technology in Jilin University in 2004. His research interests are network security and digital watermarking.

Fu Feng received the B.S. degree from the College of Computer Science and Technology, Jilin University, in China, in 2009. Currently, he is working toward the M.S. degree in Computer

Application Technology in Jilin University. His research interests are wireless sensor network QoS for multimedia transmission.

Chong-kwon Kim received the B.S. degree in industrial engineering from Seoul National University, the M.S. degree in operations research from Georgia Institute of Technology, and the Ph.D. degree in Computer Science from University of Illinois at Urbana-Champaign in 1981, 1982, and 1987, respectively. In 1987, he joined Bellcore (now, Telcodia) as an MTS and worked on Broadband ISDN and ATM QoS support. Since February 1991, he has been with Seoul National University as a Professor in the School of Electrical Engineering and Computer Science. His research interests include wireless and mobile networking, high speed network control, distributed processing, and performance evaluation.

Yin Ke-xin received the B.S. degree in Computer Software and M.S. degree in Computer Application Technology in Changchun University of Science and Technology in China, in 1998 and 2004, respectively. She also received the Ph.D. degree in the same university in 2004. Her research interests are network security and Cryptography.

Liu Yan-heng is currently a professor and PhD supervisor, his research interests are network communication, QoS of mobile IP network and network intrusion detection technology.