Combining Adaptive Filtering and IF Flows to Detect DDoS Attacks within a Router

Ruoyu Yan[1,2], Qinghua Zheng[1] and Haifei Li[3]
[1] Department of Computer Science and Technology, MOE KLINNS, Xi'an Jiaotong University
Xi'an, Shanxi, China
[e-mail: qhzheng@mail.xjtu.edu.cn]
[2] School of Information Science, Guangdong Ocean University, Zhanjiang, Guangdong, China
[e-mail: ryyan@sei.xjtu.edu.cn]
[3] Department of Computer Science, Union University, Jackson, TN, USA
[e-mail: hli@uu.edu]
*Corresponding author: Qinghua Zheng

Received February 3, 2010; revised March 17, 2010; accepted April 29, 2010; published June 30, 2010

Abstract

Traffic matrix-based anomaly detection and DDoS attacks detection in networks are research focus in the network security and traffic measurement community. In this paper, firstly, a new type of unidirectional flow called IF flow is proposed. Merits and features of IF flows are analyzed in detail and then two efficient methods are introduced in our DDoS attacks detection and evaluation scheme. The first method uses residual variance ratio to detect DDoS attacks after Recursive Least Square (RLS) filter is applied to predict IF flows. The second method uses generalized likelihood ratio (GLR) statistical test to detect DDoS attacks after a Kalman filter is applied to estimate IF flows. Based on the two complementary methods, an evaluation formula is proposed to assess the seriousness of current DDoS attacks on router ports. Furthermore, the sensitivity of three types of traffic (IF flow, input link and output link) to DDoS attacks is analyzed and compared. Experiments show that IF flow has more power to expose anomaly than the other two types of traffic. Finally, two proposed methods are compared in terms of detection rate, processing speed, etc., and also compared in detail with Principal Component Analysis (PCA) and Cumulative Sum (CUSUM) methods. The results demonstrate that adaptive filter methods have higher detection rate, lower false alarm rate and smaller detection lag time.

Keywords: Anomaly detection, distributed denial of service, Kalman filter, recursive least square, router-wide traffic analysis

A preliminary version of this paper appeared in IEEE ICCS 2008, November 19-21, Guangzhou, China. This version includes a DDoS attacks detection scheme, and a concrete analysis and comparison. The research was supported by the National High-Tech R&D Program of China (2008AA01Z131), the National Science Foundation of China(60825202, 60803079, 60633020, 60921003), the National Key Technologies R&D Program of China (2006BAK11B02, 2006BAJ07B06, 2008BAH26B02, 2009BAH51B00), the Open Project Program of the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences (20080101). We express our thanks to Dr. Junaid Khan who checked our manuscript.

DOI: 10.3837/tiis.2010.06.014

1. Introduction

Internet-based attacks can be launched anywhere in the world and any Internet-based service is a potential target for these attacks. A denial of service (DoS) attack aims to deny legitimate users to access shared services or resources [1]. When the traffic of a DoS attack comes from multiple sources, it is called a distributed denial of service (DDoS) attack. By using multiple attack sources, the power of DDoS attacks is amplified and the problem of defense becomes much more complicated. In 2004, Federal Bureau of Investigation (FBI) and Computer Security Institute (CSI) released a survey [2] on impact of global Internet security events, which shows that among all Internet-based attacks DDoS attack is the most costly one this year. As a result, how to identify and prevent DDoS attacks is crucial.

A challenge in detecting DDoS attacks is the need to deal with huge traffic in networks. Many anomaly detection approaches [3][4][5][6][7][8][9] solve the traffic processing bottleneck by aggregating traffic volume data over time instead of scrutinizing every packet. This can speed up detection and reveal important features for security management. There are many ways of aggregation [10] to meet different needs, such as aggregating traffic according to Point of Presences (POPs), links, and IP address prefixes. As for DDoS attacks detection, it is necessary to analyze a typical DDoS attack structure shown in **Fig. 1**. This structure resembles a huge funnel in which attack packets are aggregated to the target, and each router acts like a smaller funnel aggregating attack traffic from different ports to the destination port. If the attack traffic is huge, detection of the attack at destination port is not a problem. But if the attack traffic is smaller, it is always flooded by big normal traffic aggregated at egress port, so it is hardly detected. We propose a new method to cluster traffic to detect this kind of minor attacks accurately and timely, which aggregates traffic between two ports within a router. This aggregated traffic is called IF flow and it is very suitable for DDoS attacks detection as it can increase the value of the ratio of attack traffic to normal traffic compared to input links and output links.

Another challenge in identifying DDoS attacks is detection efficiency for high bandwidth networks and limited computational resources. The second major contribution of this paper is the development of a scheme to detect and evaluate DDoS attacks in a router based on IF flows, which presents an efficient way of detection compared with two existing statistical methods. To quickly and accurately detect attacks we propose two adaptive filtering-based detection methods in the scheme. The two methods can work independently under different environments according to the actual needs.

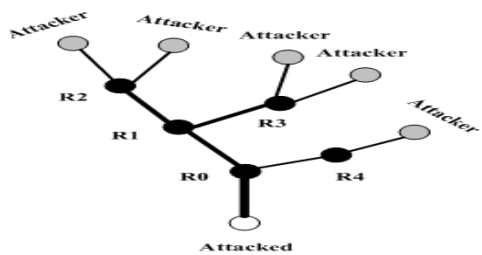


Fig. 1. DDoS attack tree

In the next section, we briefly summarize existing research on attack detection techniques, especially DoS/DDoS attacks detection in network traffic. In section 3, we define three types of traffic and discuss the merits of IF flows in detail. In section 4, we propose a DDoS attacks detection scheme. Particularly, an evaluation formula is proposed to assess how seriously a router port is under DDoS attacks. The RLS-based method and Kalman-based method to detect DDoS attacks are presented in Section 5 and Section 6 respectively. We demonstrate the effectiveness of our approach using an empirical evaluation and give some meaningful comparison results in Section 7. Section 8 gives the extension of the proposed scheme.

2. Related Work

Existing methods for DoS/DDoS attacks detection can be roughly categorized into three types according to the deployment points: 1) deployed on a link or a server,2) deployed on a router, 3) deployed on a large-scale network.

Deployed on a Link or a Server: [11] proposed a scheme called MULTOPS to detect DoS attacks by monitoring the packet rate in both the up and down links. This scheme assumes that packet rates between two hosts or subnets are proportional during normal operation. If the packet rates are significantly disproportional, it is strongly believed that a DoS attack happens. [12] proposed SYN detection to detect SYN flooding by monitoring statistical changes. The ratio of SYN packets to FIN and RST packets is used. When the random sequence is statistically non-homogeneous and the ratio changes remarkably, an attack is assumed to be detected. [13] used Kolmogorov Complexity-based algorithm to detect DDoS attacks with a high accuracy rate. This algorithm works well under the assumption that DDoS attacks always change traffic feature distribution.

Deployed on a Router: In [14] authors proposed a method to keep a history of all the legitimate IP addresses which have previously appeared in a router. When a current IP packet is coming, the history is used to decide whether a DDoS attack has happened. How to maintain and index the huge IP address database is a big problem. [15] analyzed the traffic patterns in a router and adopted a nonparametric Cumulative Sum (CUSUM) to process traffic at each input/output port. Based on this method applied to a router, a hierarchical alarm system against DDoS attacks was introduced. [16] aimed at the change of ports' input and output traffic in a core router, and employs an improved CUSUM algorithm to trace traffic statistical characteristics in real time to detect DDoS attacks. Both methods [15][16] work well under the assumption that the ratio of input traffic to output traffic is nearly a constant value. However some DDoS attacks, such as reflector attack with attackers evenly distributed in network, is hard to be detected. [6] used different level subnet prefix to aggregate traffic going to and from a router port. When traffic volume and subnet number are large, computation is a problem in real time detection. [17] proposed Renyi cross entropy to detect DDoS attacks in a router. This method can only identify whether there is an attack happening, but cannot tell exactly which ports are under attack.

Deployed on a Large-scale Network: How to detect DDoS attacks in large-scale network has become an active research area in recent years [15][18][19]. All these papers have adopted distributed detection techniques to detect DDoS attacks. Specifically, distributed detection software is deployed and running in each router. Later on, each router sends its intermediate detection results to a control center. Finally, a general result is computed by data fusion. On the other hand, network-wide anomaly detection based on traffic matrix is a novel method for detecting volume anomalies [7][8][9]. Unlike a hierarchical structure, this kind of method can accomplish anomaly detection in large-scale networks in three steps. The first step is to build

traffic matrix relationship between origin-destination (OD) flows and input links. The second step is to estimate traffic matrix by Principal Component Analysis (PCA) or Kalman filter. The third step is to build a normal profile using a statistical process. Any anomalies from potential attackers compared with the normal profile are regarded as strong indications of an attack. Practically, obtaining OD flows in real time is not easy, which makes these methods hard to be deployed in a true network environment.

From the view point of detection techniques, generally the methods mentioned above can be classified into two groups. The first group is called DoS-attack-specific detection, which is based on the special features of DoS/DDoS attacks, such as [11][12[13][14]. The second group is called anomaly detection, which models the behavior of normal traffic, and then reports any anomalies, such as [7][6][7][8][9][15][16]. Anomaly detection has become a major focus of research, due to its ability to detect new attacks, including DDoS attacks. But anomaly detection techniques are facing a dilemma of how to choose a tradeoff between processing speed and detection accuracy. For these reasons, we propose two adaptive filtering based anomaly detection algorithms to solve the dilemma. The two detection algorithms have good accuracy and fast speed, and are complementary in their applications.

The most similar work with ours is that of [9], however there are some remarkable differences. Firstly, we propose Kalman filter and RLS filter to detect and evaluate DDoS attacks in different application environments whereas [9] mainly compares four different Kalman filter based statistical methods. Furthermore, we focus on how to detect and evaluate DDoS attacks in a router whereas work in [9] detects anomalies in OD flows. Besides, we compare Kalman filter method with RLS filter method, CUSUM method and PCA method in many aspects, such as detection rate, processing speed, detection lag time. The sensitivity of three types of traffic (IF flow, input link, output link) to DDoS attacks is also compared.

3. Merits of Using IF Flows to Detect DDoS Attacks

To facilitate the discussion, three types of flow traffic are defined below.

IF flow: IF stands for internal flow in a router, which is defined as a group of packets traveling from one port to another different port in a router per unit time. It is assumed that packets traveling from one port to the same port are very sparse, so they are not considered.

Input link: A group of packets entering a router from one port per unit time.

Output link: A group of packets leaving a router from one port per unit time.

Port serial numbers are used to mark IF flow, input link and output link like in **Fig. 2**. For example, a group of packets traveling from port A to port D per unit time are marked as IF flow A-D. The traffic actually observed on input links (or output links) arises from the superposition of IF flows within a router, which can be seen in **Fig. 2**. The relationship between input links and IF flows can be concisely captured in the routing matrix H. The matrix H has size (input links count) ×(IF flows count), where $H_{ij} = 1$ if IF flow j traverses input link i, and is zero otherwise. Then the vector of traffic counts on input links (Y) is related to the vector of traffic counts on IF flow (X) by $Y = HX$ [20]. This makes it possible to build a traffic state-space model subsequently. Note that the relationship between output links and IF flows can also be built by another routing matrix.

Much of the work in anomaly detection has focused on single-link traffic data. A router-wide view of traffic enables detection of anomalies that may be ignored in individual link traffic. IF flows are easy to collect than OD flows because consideration of packet routing between multiple routers is not needed.

Assume there is a router with n ports, then it can produce $n^2 - n$ IF flows by its definition; For example, a router with 5 ports shown in **Fig. 2** can produce 20 IF flows. To simplify the discussion, a few assumptions are made according to the characteristics of DDoS attack path. First, among IF flows shown in **Fig. 2**, two of them are anomalous which are IF flow A-D and B-D. Second, traffic count on each input links or output links is 1 and traffic count on each IF flow is 1/4 on average. Third, anomaly traffic count is 1/10 on input links A and B. From the assumptions it is easy to know that anomaly traffic accounts for 10% of all traffic on input links A and B, and 20% on output links D, and 40% on IF flow A-D and B-D. Therefore using IF flows to detect anomalies is more effective than using input links and output links.

Many attackers try to distribute their DDoS attack traffic evenly in a large-scale network in order to hide attack behaviors and avoid being spotted at an early time. Schemes deployed on a single link, such as in many anomaly-based IDS systems, are hard to spot the attacks timely. In contrast, IF flows based method can find this kind of attack early, because IF flows increase value of the ratio of attack traffic to normal traffic. In addition, IF flows expose the ports from which the attack traffic is coming and to which they are aggregated. This port information is very useful for attacks defense with proper measures.

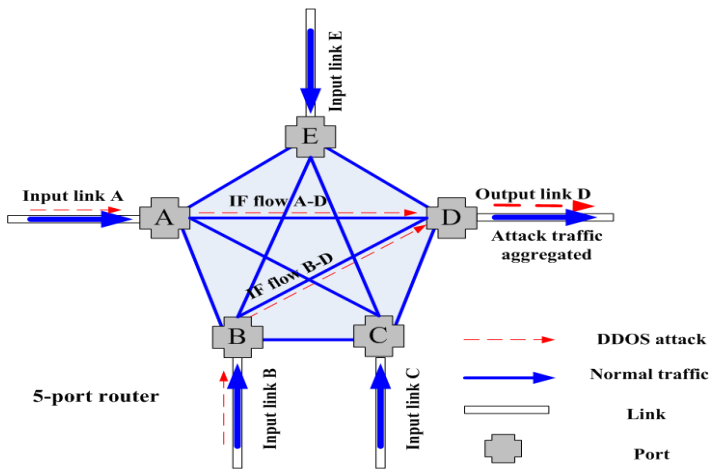


Fig. 2. DDoS attack emulation

4. DDoS Attacks Detection Scheme

According to the detailed analysis mention above and practical application needs, we propose a DDoS attacks detection and evaluation scheme based on RLS prediction detection and Kalman estimation detection. The scheme is shown in **Fig. 3**.

Traffic Collection: At present there is no tool available to obtain IF flows directly. Simple network management protocol (SNMP) can only be used to collect ports' ingress and egress traffic statistics in router. Although it is possible to obtain IF flows by analyzing packet routing, one must know routing table and monitor all packets in router. This approach consumes a significant amount of time and resources, and faces the trouble of routing table updating in real time. Fortunately, Netflow records created by Netflow cache [21] in a router can be used to achieve this goal. NetFlow, proposed by Cisco Company, is based on a flow concept. Flow is a unidirectional stream of packets with five tuples: source IP, destination IP, source port, destination port and layer 3 protocol type. After configuring a router to open Netflow cache, Netflow entries are then created at every time bin and encapsulated in UDP

packets directed to a traffic statistical analysis server. However Cisco Netflow performance analysis white paper [22] indicates that the opened Netflow function increases CPU cost a little. According to the different count of Netflow records existed in router cache, CPU used by Netflow function is 7%～23% on average. To make the scheme more flexible and robust, Kalman-based method to reduce opening Netflow function in a router is proposed.

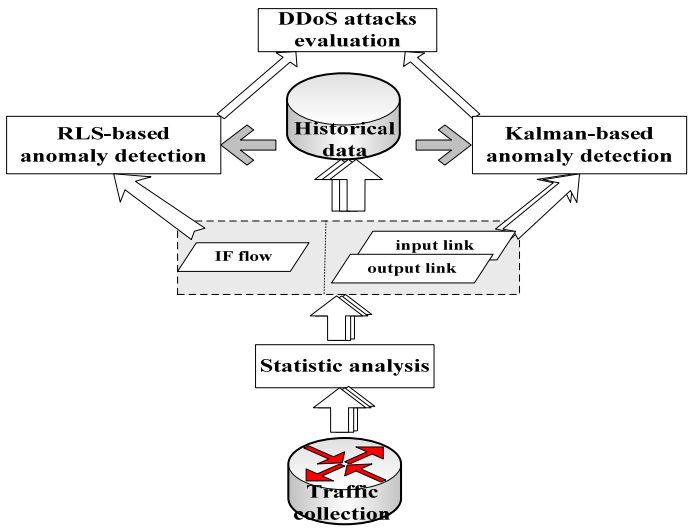


Fig. 3. DDoS attacks detection scheme

Statistical Analysis: A traffic statistical analysis server receives UDP packets at a certain UDP port. Then it unpacks the packets to extract Netflow records and stores them in a database. Because every Netflow record includes input and output fields, by using simple SQL statements, it is possible to calculate byte count, packet count and flow count of input links, output links and IF flows per unit time.

Historical Data: All historical data, which include previous Netflow records and statistical analysis results, are stored in a database for real time detection. For example, historical IF flow is needed to calculate threshold for judging anomaly before RLS-based detection method is applied in real time; history of IF flow and input link(or output link) is needed to estimate iteration parameters before Kalman-based detection method is applied in real time.

RLS-based Anomaly Detection: The normal historical IF flow is needed to calculate detection threshold beforehand. When a new IF flow statistical value arrives, this module calculates the prediction error between the new value and its RLS prediction value computed at previous time bin in real time. Then a ratio of prediction error variance in a history window to prediction error variance in detection window is used as a statistic to detect anomaly. This method has higher detection accuracy but with relatively lower processing speed. Detailed procedures are presented in section 5.

Kalman-based Anomaly Detection: Before Kalman filter is applied to estimate IF flow matrix, historical IF flows and input links (or output links) are needed to estimate iteration parameters in discrete Kalman equations. When a new input link (or output link) statistical value arrives, this module estimates IF flow value and calculates the prediction residual timely. Then a GLR statistical test method is used to detect anomaly in real time. This method has faster processing speed but with relatively lower detection accuracy. Detailed procedures are presented in section 6.

DDoS Attacks Evaluation: DDoS attacks evaluation is a method to judge how seriously each

router port is under DDoS attacks. The evaluation mechanism has fully utilized the funnel structure of DDoS attacks. Two factors that affect the evaluation index are considered. Factor one is how many IF flows going to a port are detected as anomaly at one time. A many number of anomalous IF flows going to a port imply that the port is under DDoS attack more seriously. Factor two is how large is the anomalous IF flow's traffic volume. A larger value of the ratio of anomalous IF flow's traffic volume to the port's out traffic volume implies that the port is under DDoS attack more seriously. The evaluation equation is given below.

$$E_i(t)=\begin{cases}\alpha\times\dfrac{\sum_{j=1}^{n}F_{ji}(t)}{n-1}+\beta\times\sum_{j=1}^{n}W_{ji}(t)F_{ji}(t) & n\le 5 \ \ and \ \ \sum_{j=1}^{n}F_{ji}(t)>0\\ \alpha\times\dfrac{1}{1+\exp(\lceil n/2\rceil-\sum_{j=1}^{n}F_{ji}(t))}+\beta\times\sum_{j=1}^{n}W_{ji}(t)F_{ji}(t) & n>5 \ \ and \ \ \sum_{j=1}^{n}F_{ji}(t)>0\\ 0 & \sum_{j=1}^{n}F_{ji}(t)=0\end{cases} \quad (1)$$

Where $\alpha+\beta=1$, n is the count of active router ports with traversing packets. i and j denote port number. $E_i(t)$ denotes the intensity of DDoS attacks against port i at time t, namely evaluation index. α and β denote the weight of factor one and the weight of factor two respectively in evaluation index. For example, if $\alpha>0.5$, it means anomalous IF flow count has been considered more important than the volume of anomaly traffic in computation of evaluation index. When $j\neq i$, $F_{ji}(t)$ is anomalous value of IF flow $j-i$ at time t, where $F_{ji}(t)=1$ if IF flow $j-i$ is abnormal, and is zero otherwise; $W_{ji}(t)$ is value of the ratio of IF flow $j-i$ traffic count to port #i out traffic count (namely output link #i traffic count) at time t. When $j=i$, then $F_{ji}(t)=0$ and $W_{ji}(t)=0$ in terms of IF flow definition. Because the same count of abnormal IF flows impacts bigger on router with a small number of ports than with a large number of ports. So in practice if a router has port number not more than five, we use a linear function with big slope to denote the contribution of the count of abnormal IF flows, otherwise sigmoid function is used. Sigmoid function is a good threshold function, and its form characteristic is that its front part and back part have smaller slope, but middle part has steep slope. Sigmoid function captures the situation well: smaller count of abnormal IF flows affects lesser on the port; when the count is larger to an extent, the effect on port increases very fast, but when the count further increases, the impact of newly added abnormal IF flows is relatively limited.

5. RLS-based Anomaly Detection

5.1 Detection Algorithm Model

The detection algorithm model based on RLS is shown in **Fig. 4**. It is applied to three types of traffic in the following experiments: IF flows, input links, output links. We take IF flows as an example.

1. IF flow statistic module. This module is the same as the statistic analysis module shown in **Fig. 3**. It is used to sum up byte count, packet count and flow count of IF flows as statistics.

2. RLS prediction module. According to different needs, IF Flow statistics are selected and then predicted by RLS at every time bin. Detailed steps of prediction are given in section 5.2.
3. Prediction error module. When a new IF Flow statistical value arrives at some time, prediction error is calculated. Predication error is the difference between newly arrived value and its predicted one computed previously.
4. Variance ratio statistic detection module. It deals with prediction error in detection window and historical window. Detailed steps of detection are given in section 5.3.
5. Judge anomaly module. Alarm threshold is first set by the variance analysis of historical traffic. If an arrived value goes beyond alarm threshold, anomaly is deemed to exist.

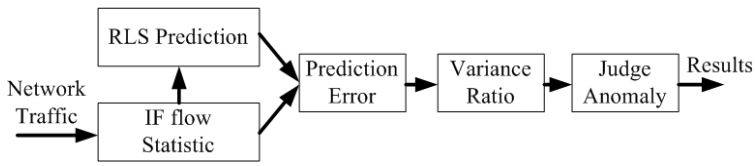


Fig. 4. Traffic anomaly detection model

5.2 Network Traffic Prediction Based on RLS

Instead of Auto-Regressive (AR) model, RLS is selected as the prediction algorithm for the following reasons: 1) AR model must update weights periodically in order to deal with non-stationary stochastic time series, which increases computation load. The situation becomes worse when facing multiple statistics.2) Before prediction, AR model must fit a section of series to get weights and the length of fitted series affects the prediction precision and calculation speed.3) RLS is adaptive to non-stationary stochastic time series, with faster speed and lesser memory when using with a smaller order, such as 2.

RLS algorithm is a kind of Kalman filter in nature [23], which exactly meets least square criterion. RLS is mainly used to filter signal noise, but also can predict signal. Literature [24] points out that with the increase of prediction step, the prediction precision will decrease fast; hence RLS is used to predict traffic in one step.

Suppose $d(n)$ called desired response is our expected signal at time n. We attempt to predict the desired network traffic $d(n) = x(n)$ by use of RLS filter. The filter coefficients at time n (namely weight vector with N dimensions) are set as $W(n) = [w_0(n) \quad w_1(n) \quad \cdots \quad w_{N-1}(n)]^T \quad n = 1,2,\cdots,k$. When historical traffic vector with N dimensions at time n is known and is $X_N(n) = [x(n-1) \quad x(n-2) \quad \cdots \quad x(n-N)]^T$, $d(n)$ can be predicted a priori by $\hat{d}(n) = X_N^T(n-1)W(n-1)$. When weight vector dimension N becomes bigger and, more historical information is used, the prediction result approximates the pure signal. But a larger N also leads to more computational work. In the experiment N is set as 2. The initial Weight vector can be set as any smaller values, such as 0, because the filter recursion process can update weight vector iteratively.

The following is the algorithm for prediction:

Initialization: $W(0) = X_N(0) = 0$, $T(0) = \delta^{-1} I$, where δ is small positive constant, I is the N-by-N identity matrix.

For each unit time, $n = 1,2,...$

1. Read input values: $d(n), X_N(n)$
2. Update prediction error: $\xi(n \mid n-1) = d(n) - X_N^T(n-1)W(n-1)$ (2)
3. Update information gain vector: $G(n) = \dfrac{T(n-1)X_N(n)}{\lambda + X_N^T(n)T(n-1)X_N(n)}$ (3)
4. Update weight vector: $W(n) = W(n-1) + G(n)\xi(n \mid n-1)$ (4)
5. Update inverse of correlation matrix: $T(n) = \dfrac{1}{\lambda}\left[T(n-1) - G(n)X_N^T(n)T(n-1)\right]$ (5)

It can be seen that weight vector is corrected by gain and error vector. Equation (3), (5) update the value of the gain vector itself. The inverse of the correlation matrix is replaced at each step by a simple scalar division. Note λ is forgetting factor. The smaller λ is, the smaller contribution of previous samples. This makes the filter more sensitive to recent samples, which means more fluctuations in the filter coefficients. Generally λ is set as 0.99.

5.3 Variance Ratio Statistic Detection Algorithm

Historical traffic is obtained beforehand by monitoring the normal network. After its prediction, prediction error variance is calculated. The variance captures statistical characteristics of normal historical traffic. When in real time detection, 'sliding window' variance ratio detection algorithm shown in **Fig. 5** is introduced to identify anomaly. Two sliding windows are used in the algorithm: history window (HisWin) and detection window (DetWin). Both of them are sliding in real time.

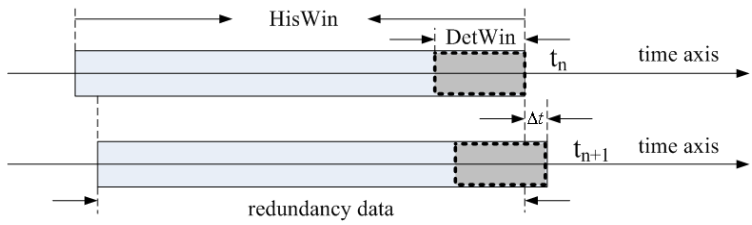


Fig. 5. Sliding window's variance ratio detection

At first error variance $DetV_n$ of data in detection window $(t_n - DetWin, t_n)$ and error variance $HisV_n$ of data in history window $(t_n - HisWin, t_n)$ at time t_n are computed. Then variable $ratio_n = (DetV_n / HisV_n)^2$ is computed to denote the departure of data in detection window from data in history window at time t_n. If there is an anomaly adding to detection window at present, the $ratio$ value would increase remarkably. As a result, $ratio$ changes can be used to find the anomaly. There are three parameters in the detection algorithm:

1. Detection window size. It is ideal if the size of detection window is equal to the lasting time of possible anomaly. However, it is impossible to know in advance the lasting time of anomaly traffic to be detected. However, it is possible to know the lasting time of all historical anomaly traffic. Therefore, the average lasting time of all historical anomaly traffic is selected as a detection window size.
2. History window size. A big history window size leads to a more accurate detection results. However, too large history window size increases the cost of system's storage and computation. Here history window size is set as 100.
3. Anomaly threshold. Anomaly threshold is $ratio_{threshold} = \bar{x} + m\sigma$. Where $\bar{x}$ and

σ denote average value and standard deviation of *ratio* which is computed from normal historical traffic collected beforehand, and m is a smaller positive value between 1 to 4. When $ratio_n > ratio_{threshold}$, an anomaly is assumed to be detected at time n.

6. Kalman-based Anomaly Detection

RLS-based method can only predict one IF flow at one time. However Kalman-based method can process traffic matrix as a whole and all IF flows can be estimated simultaneously. As a result, Kalman-based method has an advantage in terms of speed. Moreover, in real time detection, Kalman-based method does not need to collect Netflow records. It means that the method has little effects on router performance.

6.1 Building Traffic State-space Model

In **Fig. 2** the traffic actually observed on input links (similarly on output links) arises from the superposition of IF flows within a router. In order to reflect the inaccuracy of data collection, equation (6) is used to express the relationship between input links (or output links) and IF flows.

$$Y_t = \mathrm{H}_t X_t + V_t \tag{6}$$

where Y_t denotes an input link traffic vector (observation vector), and X_t denotes an IF flow vector (hidden variable), and H_t denotes an internal routing matrix, where $H_{ij} = 1$ if IF flow *j* traverses an input link #*i*, and is zero otherwise. As the traffic collecting device may cause measurement errors, stochastic variable V_t is used to capture this error.

Although prediction model can have any structure, and noise process can have any distribution, the combination of linear stochastic prediction model and Gaussian noise has been successfully applied to solve many problems recently. Hence IF flow is treated here as network state X_t, and a linear equation is constructed as follows to build a prediction model to correlate X_{t+1} and X_t.

$$X_{t+1} = \Phi_t X_t + W_t \tag{7}$$

where state transition matrix Φ_t captures temporal and spatial correlations, and W_t is a noise stochastic process which denotes the randomness and unpredictability existed in IF flows. For a single IF flow, the diagonal elements of Φ_t capture the temporal correlations appearing in the IF flow's evolution. The non-diagonal elements of Φ_t describe the dependency of one IF flow on another, thus capturing any spatial correlations among the IF flows (if and when they exist). This model follows the form of a typical linear time-invariant dynamical system.

The combination of equations (6) and (7) becomes a linear state space dynamic system, given by:

$$\begin{cases} X_{t+1} = \Phi_t X_t + W_t \\ Y_t = \mathrm{H}_t X_t + V_t \end{cases} \tag{8}$$

where state noise W_t and measurement noise V_t are uncorrelated, zero-mean white-noise processes and with covariance matrices Q_t and R_t respectively. If the dynamic system model

is known, the optimal estimation of real network state X_{t+1} is possible when a series of observations $\{Y_1,...,Y_{t+1}\}$ are given. Kalman filter is a classical method to solve this problem.

6.2 Discrete Kalman Filtering Equations

The Kalman filtering equations [23] applied to discrete time-varying system are listed below, which includes prediction procedure given by equations (9) and update procedure in equations (10). Some important variables used in the equations are explained in [23]. When initial conditions, $\hat{X}_0^0 = E[X_0]$ and error covariance matrix $P_0^0 = E[(\hat{X}_0^0 - X_0)(\hat{X}_0^0 - X_0)^T]$, are known, the system state $\hat{X}_{t+1}^{t+1}$ can be estimated iteratively by equations (9) and (10).

$$\begin{cases} \hat{X}_{t+1}^{t} = \Phi_t \hat{X}_t^t \\ P_{t+1}^t = \Phi_t P_t^t (\Phi_t)^T + Q_t \end{cases} \tag{9}$$

$$\begin{cases} \hat{X}_{t+1}^{t+1} = \hat{X}_{t+1}^t + K_{t+1}[Y_{t+1} - \mathrm{H}_{t+1}\hat{X}_{t+1}^t] \\ P_{t+1}^{t+1} = (I - K_{t+1}\mathrm{H}_{t+1})P_{t+1}^t \\ K_{t+1} = P_{t+1}^t(\mathrm{H}_{t+1})^T[\mathrm{H}_t P_{t+1}^t(\mathrm{H}_{t+1})^T + R_{t+1}]^{-1} \end{cases} \tag{10}$$

6.3 Using Expectation Maximization (EM) Algorithm to Estimate $\{\Phi, Q, R\}$

Equations (9) and (10) are general descriptions of Kalman filter under non-stationary state. The detection results are rarely affected if the $\{\Phi, Q, R\}$ is constant and are not calibrated for even about one week. In order to improve detection speed and decrease data collection in actual applications, $\{H, \Phi, Q, R\}$ is assumed to be constant. So their time subscripts in equations (9) and (10) are removed in the following experiments.

$\{H, \Phi, Q, R\}$ should be known before using Kalman filter to estimate traffic matrix. The routing matrix H is an already given constant in the light of traffic relationship between IF flows and input links (or output links) in router. The remaining system parameter $\theta = \{\Phi, Q, R\}$ needs to be calculated. We use EM algorithm to estimate it. EM computes maximum likelihood estimation of θ recursively.

Suppose, the system state is observable in equation (8), θ can be estimated by maximum likelihood estimation if $Y = [Y_0 Y_1 ... Y_n]$ and $X = [X_0 X_1 ... X_n]$ are known. The maximizing equation is $\ln L(X, Y, \theta) = -\frac{n}{2}\log|Q| - \frac{1}{2}\sum_{k=1}^{n}(X_k - \Phi X_{k-1})^T Q^{-1}(X_k - \Phi X_{k-1}) - \frac{n}{2}\log|R|$

$$-\frac{1}{2}\sum_{k=1}^{n}(Y_k - \mathrm{H}X_k)^T R^{-1}(Y_k - \mathrm{H}X_k) + CONSTANT$$

The system parameters are derived as follows [25][26].

$$\hat{\Phi} = BA^{-1}, \tag{11}$$

$$\hat{Q} = n^{-1}(C - B\Phi^T - \Phi B^T + \Phi A \Phi^T), \tag{12}$$

$$\hat{R} = n^{-1}\sum_{t=1}^{n}[(Y_t - \mathrm{H}X_t^n)(Y_t - \mathrm{H}X_t^n)^T + \mathrm{H}P_t^n\mathrm{H}^T]. \tag{13}$$

In the above equations, A,B and C are defined as follows.

$$A = \sum_{t=1}^{n} [P_{t-1}^{n} + X_{t-1}^{n} (X_{t-1}^{n})^{T}], \tag{14}$$

$$B = \sum_{t=1}^{n} [P_{t,t-1}^{n} + X_{t}^{n} (X_{t-1}^{n})^{T}], \tag{15}$$

$$C = \sum_{t=1}^{n} [P_{t}^{n} + X_{t}^{n} (X_{t}^{n})^{T}]. \tag{16}$$

Constant-interval Kalman smoothing filter is used to compute needed parameters $\{A, B, C\}$. This filtering procedure includes standard Kalman filter forward recursion and Kalman filter backward recursion [25][26][27]. Forward recursion procedure includes previous equations (9) and (10), but parameters $\{H, \Phi, Q, R\}$ are not dependent on time.

Kalman filter backward recursion includes:

For t=n,n-1,…,1, $X_{t-1}^{n} = X_{t-1}^{t-1} + J_t (X_t^{n} - X_t^{t-1})$, where $J_t = P_{t-1}^{t-1} \Phi^{T} (P_t^{t-1})^{-1}$ (17)

$$P_{t-1}^{n} = P_{t-1}^{t-1} + J_t (P_t^{n} - P_t^{t-1}) J_t^{T} \tag{18}$$

For t=n,n-1,…,2,

$P_{t-1,t-2}^{n} = P_{t-1}^{t-1} J_{t-1}^{T} + J_t (P_{t,t-1}^{n} - \Phi P_{t-1}^{t-1}) J_{t-1}^{T}$, where $P_{n,n-1}^{n} = (I - K_n H) \Phi P_{n-1}^{n-1}$ (19)

Note that some values such as X_n^{n}、P_n^{n}、P_n^{n-1} and P_{n-1}^{n-1} in backward recursion are initialized by corresponding final values computed from forward recursion. To sum up, **Fig. 6** shows a flow chart displaying how EM algorithm estimates system parameter θ. Φ, Q and R can be initialized as unit matrix. The larger the recursion times, the higher estimation precision is. But time cost is much higher and convergence rate is slower with the increase of recursion times. Actually instead of estimation precision, a suitable recursion times is always used as a condition to end estimation.

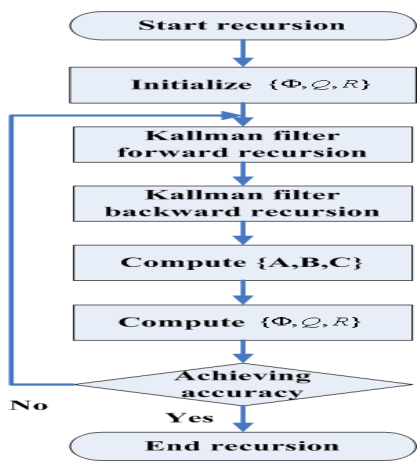


Fig. 6. Flow chart of EM estimation of $\{\Phi, Q, R\}$

6.4 Generalized Likelihood Ratio (GLR)-Based Statistical Test

One step prediction value and estimation value can be gotten directly from Kalman filter equations. innovation (δ_{t+1}) and residual (η_{t+1}) of IF flow are computed as follows.

$$\delta_{t+1} = X_{t+1} - \hat{X}_{t+1}^{t} \tag{20}$$

$$\eta_{t+1} = \hat{X}_{t+1}^{t+1} - \hat{X}_{t+1}^{t} = K_{t+1} e_{t+1} \tag{21}$$

δ_{t+1} can be used to detect anomaly in IF flow traffic, but calculating δ_{t+1} is very complicated if X_{t+1} is unknown. Instead of δ_{t+1}, residual η_{t+1} is used as statistics to detect anomaly through GLR test given in [28]. GLR test can make best estimation under unknown mean and variance of time series. In the next section the Kalman-based method is compared with RLS-based method.

7. Comparison and Analysis

7.1 Measurement Data Used

To validate any anomaly detection method, one common approach is to collect live data in the form of a packet or flow level trace, but operators must examine these data and "mark" anomaly event. It is hard to label or mark a trace, because operators can make mistakes by either missing an anomaly or generating a false positive. At the same time such live data contain a limited number of anomaly events whose parameters cannot be varied. Therefore, it is necessary to create synthetic attacks as test samples. The advantage of this approach is that the parameters of an attack can be carefully controlled.

The method of creating synthetic anomaly described in [9] in detail is used to create DDoS attack traffic as experiment data. Concrete procedure is as follows.

1. Collect a week of NetFlow traffic in a five-port router at intervals of one minute in Xi'an JiaoTong University. IF flows traffic can be aggregated by using input field and output field in NetFlow records. Packet counts of IF flows are calculated as measurement metric and used in the following experiments.
2. Use Daubechies-5 discrete wavelet transform to extract the long-term statistical trend from the selected IF flows. The goal is to capture the diurnal pattern by smoothing the original signal.
3. Add to the smoothed IF flow a zero mean Gaussian noise whose variance is computed as follows. Take the first 5 detailed signals from wavelet transform, and compute the variance of the sum of the 5 detailed signals. A background IF flow traffic is created.
4. Randomly select values of four parameters in **Table** 1 to characterize DDoS attacks. Add the anomaly on top of the background traffic.
5. Use the created IF flow traffic to infer input links and output links traffic according to internal traffic matrix relationship within a router.

Although most DDoS attacks last between 5 to 30 minutes [29], there are some outliers that last less than 1 minute or more than one day. Here the attack lasting time is selected between 1 and 30 minutes. In **Table** 1, δ is a multiplicative factor which is multiple of 0.1, and multiplied by the IF flow baseline traffic to generate the attack traffic load. δ actually denotes ratio of DDoS attack traffic to normal IF flow traffic. For each δ 40 DDoS attacks are generated starting at different time, and each attack affects 2.5 IF flows on average. (*Src*,*Dst*) refers to DDoS attack coming from *Src* ports and leaving from *Dst* ports. Here $Dst=1$ indicates DDoS attack traffics are only aggregated to one egress port. Ramp and Exponential

are shape functions, detailed description about which can be found in literature [9].

Table 1. DDoS attack description parameters

Parameter	Duration(min)	Attack intensity	Num (Src,Dst)	Shape
Possible values	1~30	$0.1 \le \delta \le 1$	(1, 1)～ (4, 1)	Ramp, Exponential

7.2 Validation of DDoS Attacks Detection and Evaluation Methods

Validation of RLS-based method: **Fig. 7** shows results of RLS-based method applied to a synthetic IF flow traffic measured by packet count. The synthetic IF flow is created by using the method mentioned above. In the first sub figure synthetic IF flow with DDoS attack is marked in blue solid curve and its predicted traffic is marked in red dash-dot curve. The two vertical dotted lines mark the start and end time of a DDoS attack. The second sub figure shows the results of variance ratio statistical method directly applied to original traffic. The third sub figure shows the prediction error at different times. It can be validated that prediction error is normally distributed with mean zero. The final sub figure is detection results of variance ratio statistic method applied to prediction error. In sub figure 2 and 4 red horizontal line is the threshold when m is 3; blue curve is variance ratio. From sub figure 2 and 4 it can be seen that prediction error has more powerful ability than original traffic to detect anomaly. Not only that it can make lesser mistakes, but also it can detect DDoS attacks at the very start and is able to tell its duration precisely.

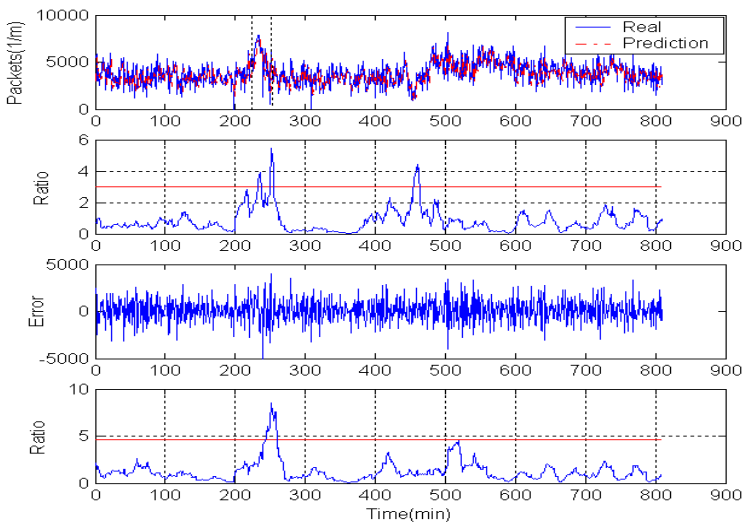


Fig. 7. RLS-based method to detect synthetic IF flow

Validation of Kalman-based method: **Fig. 8** shows results of Kalman-based method applied to an original IF flow traffic measured by flow count. The upper sub figure shows the original IF flow (blue solid curve) and its estimated IF flow (red dash-dot curve). It can be seen that the estimated results can capture the trend of original traffic's changes. The middle sub figure shows the residual at different times. The bottom sub figure shows the results of GLR test [28] applied to determine anomaly. The size of time window is set as 30 in GLR test. Red horizontal line and black dashed horizontal line show control limit for α value of 0.05 and 0.005 respectively. During the time bin from 115 to 179 there is violent vibration taking place

in the IF flow, besides there are some serious burr phenomena scattered in the traffic. In the middle of IF flow time series traffic count is very small because of night time. All these phenomena are illustrated properly in bottom sub figure.

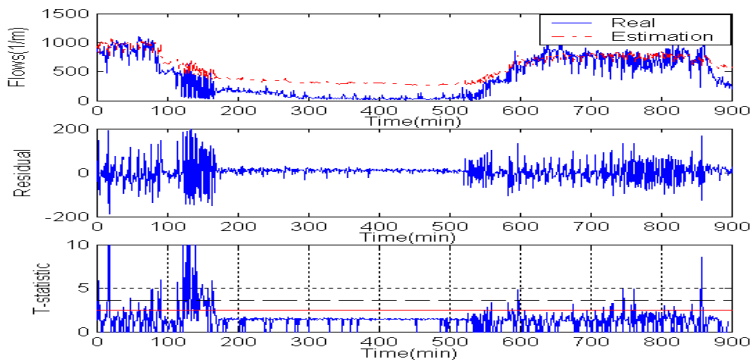


Fig. 8. Kalman-based method to detect original IF flow

Validation of DDoS attacks Evaluation: **Fig. 9** show the DDoS attack against port #4 happened around 350 time bin which is carefully selected and validated. Here RLS-based method is used to detect anomaly and threshold is set as m=3. **Fig. 10** shows evaluation results of DDoS attacks against port #4. In the figure port #4 is regarded in healthy state (namely under no DDoS attacks and attack index is zero) during all the time except for some sharper spikes caused by DDoS attacks or normal traffic vibration. Compared with **Fig. 9** it is not difficult to find that the time when the highest index values have appeared is right the same time when DDoS attack has happened. The method shows effective in evaluating severity of DDoS attacks against router port.

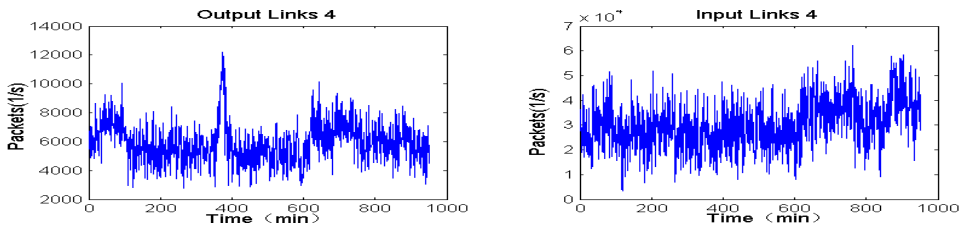


Fig. 9. output link #4 and input link #4

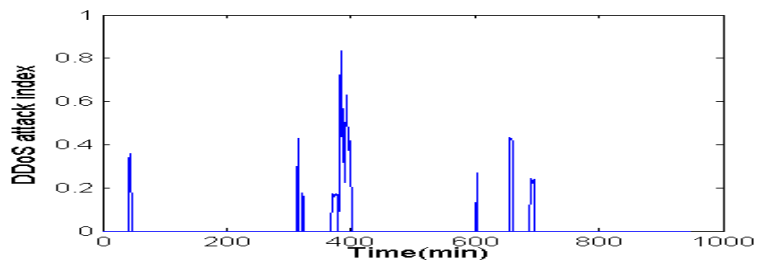


Fig. 10. Evaluation results of DDoS attacks against port #4

7.3 Results Comparison of Three Types of Traffic

Within each type of traffic, for each value of the threshold, the entire traffic matrix (thus traversing all anomalies and non-anomalies) is examined. One false positive percentage and

one false negative percentage for each threshold configuration of a scheme are computed. The performance of the method applied to three types of flows is depicted in Receiver Operation Characteristic (ROC) curves. The ROC curve is the plot of true positive ratio (TPR) against False Positive Ratio (FPR). TPR is the fraction of DDoS attacks traffic correctly classified as DDoS attacks. The false negative ratio (FNR) is the fraction of DDoS attacks traffic wrongly classified as normal traffic. FPR is the fraction of normal traffic wrongly classified as DDoS attacks. An algorithm is considered better if its ROC curve climbs rapidly towards the upper left corner of the graph. In **Fig. 11**, one FNR and one FPR are results of detecting synthetic anomaly traffic with the same attack intensity. In **Fig. 12**, one TPR and one FPR are detection results at the same threshold.

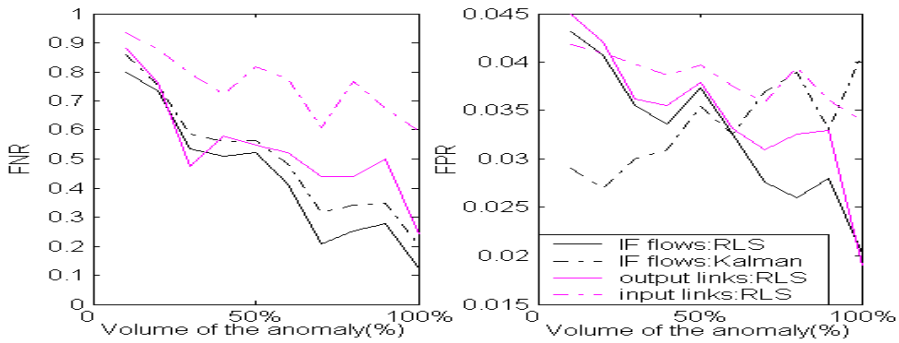


Fig. 11. FNR and FPR as a function of the attack intensity

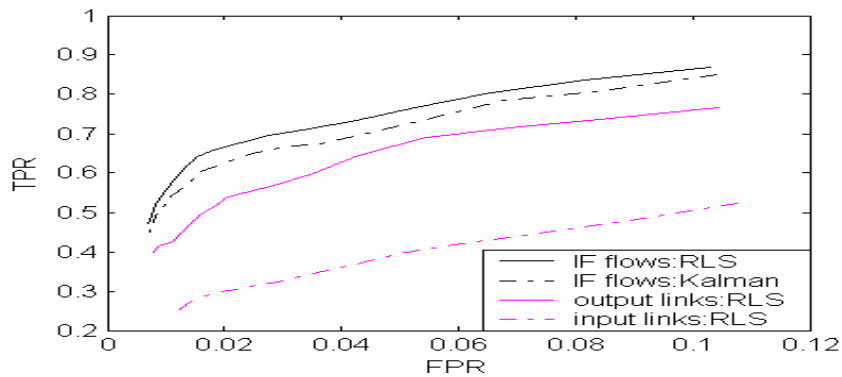


Fig. 12. ROC curves using synthetic data

The more enormous anomalies, the easier is detection, and vice versa. FNR and FPR also decrease with the increase of anomaly intensity. The same conclusion is shown in **Fig. 11** and **12**. But incidentally, in **Fig. 11** FPR of Kalman-based method is increasing with the increase of attack intensity. This is because IF flows estimated by Kalman filter can affect each other in estimating results, which causes propagation of estimation error between IF flows. All these observations given below are only within the detection results of RLS-based method.

1. In **Fig. 11**, with the decrease of attack size, the curves for FPR and FNR of the three types of flows are more convergent. It is easy to find that, when attack intensity is trailing off to a certain degree, it is hard to detect anomaly with the three types of flows.
2. With the increase of attack intensity, FPR and FNR of IF flows are smaller than that of input links and output links; FPR and FNR of output links are smaller than that of input links.

3. **Fig. 12** shows anomaly detection in IF flows more effective than in input links and output links. Clearly at the same FPR, TPR of IF flows is about 10% higher than that of output links, 35% higher than that of input links.

In summary, if IF flows can be obtained easily, detection with IF flows is a good option. However, if it is not accessible, output links are more suitable than input links for detection.

7.4 Results Comparison of RLS-based Method and Kalman- based Method

These two methods are compared in terms of detection rate, detection speed, applicability, and difficulty levels associated with realization. Both are complementary to each other, and suitable for different application environments. This makes the scheme more powerful,flexible and robust.

1. Detection rate. **Fig. 12** shows, under the same false alarm rate, detection rate of RLS-based detection in IF flows is a little higher than that of Kalman-based detection in IF flows. One reason is that Kalman-based method is lesser accurate in estimating IF flows than RLS-based method, which decreases the detection accuracy of Kalman-based method. Another reason is that the two detection methods use different statistical detection algorithm, which also causes different detection results.
2. Detection speed. Kalman-based method needs to calculate parameters recursively beforehand. RLS-based method also needs lots of historical traffic and computation to calculate thresholds. But these early work has little effect on real time detection, only the detection speed of two methods are compared in real time. A computer with a CPU of Pentium-IV 3.0GHz and memory of 752M is used to run MATLAB programs of two methods respectively for 500,000 iterations. Experiments show that Kalman-based method needs 0.41 ms to detect once in average, however RLS-based method needs 1.45 ms. Kalman-based method is faster than RLS-based method. Of course, as a matter of fact, two methods have enough detection speed to detect anomaly in real time.
3. Applicability. Before using RLS-based method, it is necessary to collect previous IF flows in real time, which leads to a burden on a router and affect its normal operation. However Kalman-based method only needs IF flows to estimate iteration parameters beforehand. Once the iteration parameters are estimated, there is no need of IF flows to calibrate the iteration parameters for a long period of time (such as one week) and only input links or output links are needed as observation vectors. These two types of traffic can be obtained by SNMP to access Management Information Base (MIB) in a router without using Netflow cache. As a result, its overhead on router is very small. Clearly, RLS-based method is suitable for a lightly loaded router, but Kalman-based method can be applied to a heavily loaded router.
4. Difficulty levels associated with realization. There are a lot of matrix operations used in Kalman-based method, especially formulas to calculate iteration parameters, which makes the method difficult to perform in practice. Nevertheless it is easy to estimate traffic in real time and IF flow matrix can be estimated in one stroke. RLS-based method is simple in predicting traffic and easy to implement. But it can only predict one statistic at a time, and the more statistics arrived at one time, the more times RLS-based method performs. Fortunately, IF flows are independent of each other and can be obtained simultaneously, which make RLS-based method easy to predict traffic in parallel. At the same time, variance ratio test and GLR statistical test can be performed in iterative optimization, which can reduce redundant computation.

7.5 Comparison with CUSUM Detection Method

CUSUM method has a strong real time detection power. It is often used in all kinds of statistical process control. In [16], authors made some improvements on general non parametric CUSUM and used traffic ratio of input port traffic to output port traffic to detect DDoS attacks in router. However they did not give any efficiency experiment although some results shown are perfect. Before using this method in real time detection, experiments are needed to determine some parameters. For a general view of detection performance, some typical values for threshold multiplier λ and skewed ratio μ are selected respectively. λ is selected between 2 to 8 uniformly. μ is selected between 0.05 to 0.45 uniformly. Input links and output links traffic aggregated in previous experiment (namely input and output port traffic) are used as test data. The stable mean parameters of five router ports computed as in [16] are $\delta = (0.2596, 0.0965, 0.2749, 0.2154, 0.1367)$. Forgetting factor β is set as 0.01. The experiment results are a series of ROC curves shown in **Fig. 13**. Each ROC curve corresponds to each μ.

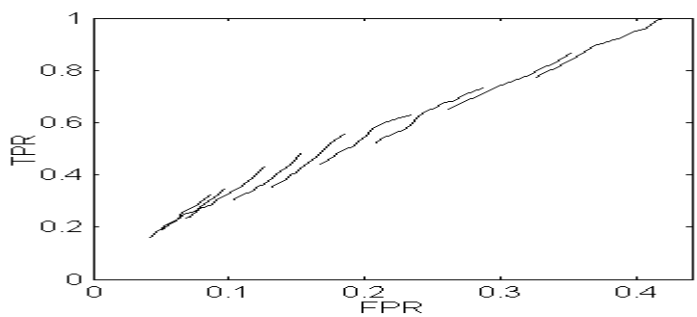


Fig. 13 A series of ROC curves of CUSUM method

We can see that all the ROC curves are almost linear with same slope. That means for different μ, the detection results is consistent as a whole, if only λ is set within wide enough range. Comparing **Fig. 13** with **Fig. 12**, it is easy to find that CUSUM method has lower detection rate than both Kalman method and RLS method when they are applied to three types of traffic. At the same detection rate, CUSUM method causes much more false alarms, making it unsuitable for practical application when attack traffic is very small. There are two main reasons which lead to poor performance of CUSUM except for small attack traffic. One reason is that although CUSUM algorithm can detect attack, but when attack is ended, the CUSUM cumulative value of the follow-up is still large enough to exceed threshold value, which results in false alarms frequently, especially when attack lasts for a long time. Another reason is that the traffic sampling time is one minute but in [16] is 100 ms. The normal traffic display stronger stationarity under Statistical multiplexing [30] for shorter sampling time. So the traffic is not very stationary because of longer sampling time. This makes it not easy to obtain a reasonable mean δ, which affects CUSUM detection performance.

7.6 Comparison with PCA Detection Method

In [8], authors used PCA based Q-statistic to detect anomaly in input links traffic. The results showed PCA based Q-statistic have strong detection performance. At first, this method use PCA to separate principal component of offline traffic matrix traces. Then the squared prediction error (SPE) is computed between original vector and predicted vector. At last, the threshold δ_α^2 for SPE at the $1-\alpha$ confidence level is set. When $SPE > \delta_\alpha^2$, the network traffic is considered abnormal.

The test data used in the experiment are created by the same man made traffic method mentioned before. The difference between them is that attack intensity δ has a much wide value from 0.1 to 2. For each δ, 100 DDoS attacks are randomly generated. RLS, Kalman and PCA methods are all applied to IF flow traffic. The comparison results of them are shown in **Fig. 14** and **Fig. 15**.

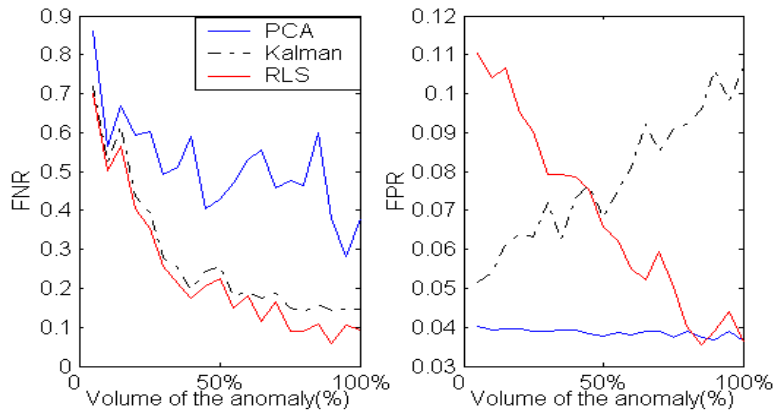


Fig. 14. FNR and FPR as a function of the attack intensity

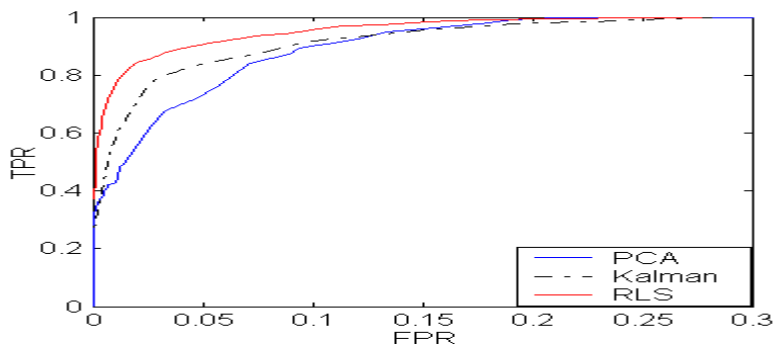


Fig. 15. ROC curves of three methods

Fig. 14 and **Fig. 15** show RLS and Kalman methods have consistent results in general compared with previous experiments. Next performance comparison and analysis in detail are given among the two methods and PCA method.

In **Fig. 14**, left subfigure shows false negative ratio of PCA is much higher than that of RLS and Kalman. With the increase of attack intensity, although the false negative ratio of PCA decreases gradually, the gap between them is increasing. Right subfigure shows the false positive ratio of PCA is lower than other methods. Moreover, it changes little when attack intensity changes. This explains why PCA method can control false alarm perfectly, but at the same time increases false negative rate, causing lower detection rate.

Fig. 15 shows when FPR is between 0～14% , detection rate of Kalman is 8% higher than that of PCA on average. When FPR exceeds 14%, detection rate of PCA exceeds that of Kalman. When FPR is between 0～19% , detection rate of RLS is 13% higher than that of PCA on average. When FPR exceeds 19%, detection rate of PCA exceeds that of RLS. At last, PCA detects all anomalies when FPR is about 20%. However RLS and Kalman detect all anomalies when FPR is about 26%. However when FPR is very low (about 0~5%), RLS and Kaman methods have much higher detection rate than PCA; when FPR is very high (about

exceeding19%), detection rate of three methods differs little. Therefore RLS-based method and Kalman-based method are considered better than PCA method. In the following the detection lag time experiment proves this too.

Detection lag time is a key indicator to judge whether a detection method is good or not. Detection lag time denotes the time when detection method detects a true anomaly minus the time when the anomaly began. A good detection method should have shorter detection lag time. When detection lag time is shorter, anomaly can be detected earlier. The lag time cumulative probability distributions of three methods are shown in **Fig. 16**. PCA method can detect about 30% of all detected anomalies without delay; RLS and Kalman methods can detect about 60%. In other words, RLS and Kalman methods can easily detect anomaly and much earlier than PCA method. Even so, the detection rate is not very high for all three methods under no detection lag. This is related to trend functions selected in synthetic attack traffic creation. Different from step function which increases attack traffic to maximum value at the beginning, ramp and exponential functions used here increase attack traffic smoothly. This smooth attack traffic makes detection method harder to find anomaly with no lag.

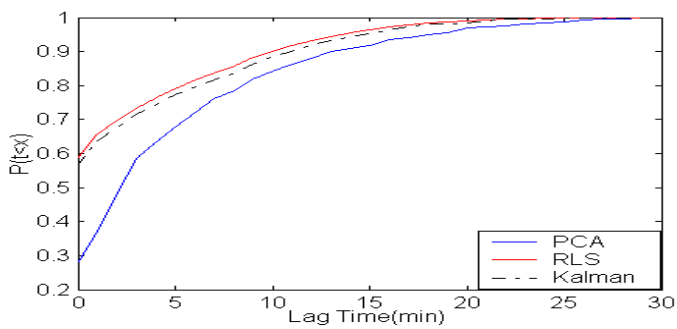


Fig. 16. Cumulative distribution function of detection lag time

8. Discussion

The scheme can be generalized to multiple core routers in large-scale network. A distributed way and co-operation between routers should be considered. A possible method to detect DDoS attacks in a large-scale network is as follows. To explain the method clearly, a 3-router network shown in **Fig. 17** is used as an example. In **Fig. 17**, number denotes router number; letter denotes router port number.

Step 1: Detect and evaluate DDoS attacks in each router to get evaluation indexes of DDoS attacks against ports and anomalous IF flows. DDoS attacks detection and evaluation results given on each router in **Fig. 17** are shown in **Table 2**. In each figure listed in **Table 2**, red arrow denotes anomalous IF flow, number denotes evaluation index of DDoS attacks against port; letter denotes router port number.

Step 2: Set a threshold of evaluation index to get attack-aggregation port and attack-entering port. For example, the threshold is set as 0.6. That means the port called attack-aggregation port, whose evaluation index is larger than 0.6, is assumed to be attacked, and the port, from which anomalous IF flow leaving attack-aggregation port enters router, is called attack-entering port. It is clear from **Table 2** that attack-aggregation port of router 1 is C, attack-entering port of router 1 is A and B; attack-aggregation port of router 2 is C, attack-entering port of router 2 is A and D; attack-aggregation port of router 3 is C, attack-entering port of router 3 is A and D.

Step 3: Merge all routers' attack-aggregation port and attack-entering port to construct a directed attack path topology. The method to determine whether attack traffic exists in the link between two routers is: if router A's connected port is attack-aggregation port and router B's connected port is attack-interring port, then attack exists in the link; furthermore, the attack direction is from router A to router B. Based on the merging method, the directed attack path topology existing in **Fig. 17** is constructed and shown in **Fig. 18**.

Step 4: Evaluate the directed attack path topology to identify DDoS attacks behavior. DDoS attacks path has some general features, such as one-way tree structure and a sole victim. If attack path topology does not meet theses features well and has poorer connectivity, lesser obvious hierarchy, there has lesser probability that DDoS attack exists in network. It is clear from **Fig. 18** that a DDoS attack exists in network shown in **Fig. 17** by the evaluation criterion.

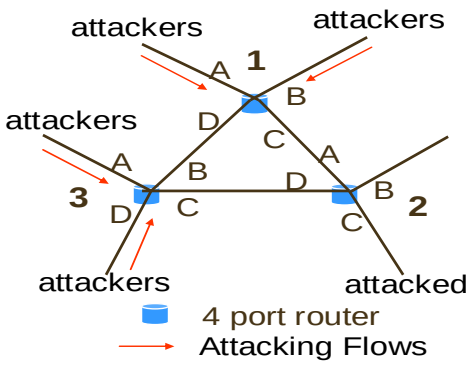


Fig. 17. 3-router network

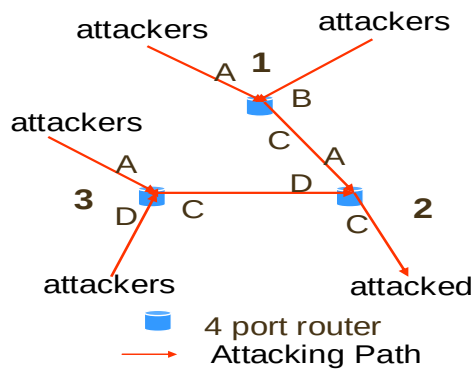
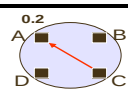
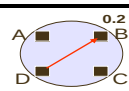
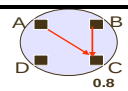
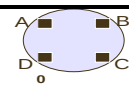
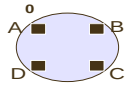
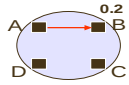
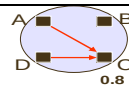
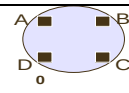
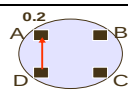
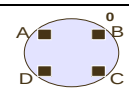
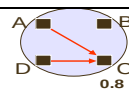
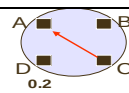


Fig. 18. The directed attack path topology

Table 2. DDoS attacks detection and evaluation results on each router

Router number	Port A	Port B	Port C	Port D
1				
2				
3				

Because the distinction between flash crowds and DDoS attacks is difficult, it seems that the scheme will have high false positive rate for "flash crowd" traffic . Some measures can be considered to solve this problem. For example, a heuristic based on the findings of [31] can be adopted. It is shown that DDoS requests came from clients widely distributed across the Internet (perhaps because DDoS attacks are more likely to be spoofed). In that light, traffic emerging from topologically clustered hosts and directed to well known destination ports (such as port 53 (DNS) or 80 (HTTP)) are classified as flash crowd. Of course, this heuristic may not always hold and as a result, some of the flash crowds may be detected as DDoS attacks in reality. On the other hand, DDoS attacks are characterized by a concentration in destination address. However flash crowds are from a dispersed set of source ports, to a concentrated set of destination addresses [32]. So entropy can be introduced in the scheme to distinguish DDoS attacks and flash crowds.

9. Conclusions

Based on the tree characteristic of DDoS attack, features of IF flows and properties of adaptive filters, this paper proposes a new scheme to detect DDoS attacks within a router. The work provides the following key contributions. 1) A new type of traffic IF flows are introduced to detect DDoS attacks, and it has been demonstrated that IF flows have more powerful ability to expose anomaly than input links and output links. 2) A detection and evaluation scheme against DDoS attacks is proposed, and it can achieve higher detection efficiency and flexibility in comparison to previous work for factors like detection rate, false alarm rate, detection lag time etc.

References

[1] V. D. Gligor, "A note on denial-of-service in operating systems," *IEEE Trans. Softw. Eng.*, vol. 10, no. 3, pp. 320-324, 1984.
[2] Computer Crime Research Center, 2004 CSI/FBI Computer Crime and Security Survey, http://www.crime-research.org/news/11.06.2004/423/
[3] P. Barford, J. Kline, D. Plonka, and A. Ron, "A Signal Analysis of Network Traffic Anomalies," in *Proc. of Internet Measurement Workshop*, 2002.
[4] S. Kim, A. Reddy, and M. Vannucci, "Detecting Traffic Anomalies at the Source through Aggregate Analysis of Packet Header Data," in *Proc. of Networking*, 2004.
[5] Tao Qin. Xiaohong Guan, Wei Li and Pinghui Wang, "Dynamic Features Measurement and Analysis for Large-Scale Networks," in *Proc. of ICC2008, CSIM workshop*, pp. 212-216, 2008.
[6] T. M. Gil, and M. Poletto, "Multops: a data-structure for bandwidth attack detection," in *Proc. of the 10th USENIX Security Symposium*, 2001.
[7] Haakon Ringberg, Augustin Soule, Jennifer Rexford, Christophe Diot, "Sensitivity of PCA for Traffic Anomaly Detection," in *Proc. of SIGMETRICS'07*,USA, pp. 109-120, June 2007.
[8] Anukool Lakhina, Mark Crovella, Christophe Diot, "Diagnosing Network-wide Traffic Anomalies," in *Proc. of SIGCOMM'04*, Portland, Oregon,USA, pp. 219-230, 2004.
[9] Augustin Soule, Kave Salamatian, Nina Taft, "Combining Filtering and Statistical Methods for Anomaly Detection," in *Proc. of Internet Measurement Conference*, pp. 331-344, 2005.
[10] A. Medina, C. Fraleigh, N. Taft, S. Bhattacharyya, C. Diot, "A Taxonomy of IP Traffic Matrices" , in *Proc. of Scalability and Traffic Control in IP Networks II*, Boston, USA, pp. 200-213, 2003.
[11] T. M. Gil and M. Poletto, "MULTOPS: A data-structure for bandwidth attack detection," in *Proc. of the 10th USENIX Security Symposium*, 2001.
[12] H. Wang, D. Zhang and K. G. Shin, "Detecting SYN flooding attacks," in *Proc. of IEEE INFOCOM*, pp. 1530-1539, 2002.
[13] Amit Kulkarni and Stephen Bush, "Detecting distributed denial-of-service attacks using kolmogorov complexity metrics," *Journal of Network and Systems Management*, vol. 14, no. 1, pp. 69-80, Mar. 2006.
[14] Peng Tao, C. Leckie and K. Ramamohanarao, "Protection from distributed denial of service attacks using history-based IP filtering," in *Proc. of ICC'03*, pp. 482-486, 2003.
[15] Yu Chen, Kai Hwang, Wei-Shinn Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Domains," *IEEE Trans. On Parallel and Distributed Systmes*, vol. 18, no. 12, pp. 1649-1662, Dec. 2007.
[16] Sun Zhi-Xin, Tang Yi-Wei, Cheng Yuan, "Router Anomaly Traffic Detection Based on Modified-CUSUM Algorithms," *Journal of Software*, vol. 16, no. 12, pp. 2117-2123, 2005.
[17] Ruoyu Yan and Qinghua Zheng, "Using Renyi Cross Entropy to Analyze Traffic Matrix and Detect DDoS attack", *Information Technology Journal*, vol. 8, no. 8, pp. 1180-1188, 2009.
[18] Krishan Kumar, R.C Joshi, Kuldip Singh, "A Distributed Approach using Entropy to Detect DDoS attacks in ISP Domain," in *Proc. of International Conference on Signal Processing, Communications and Networking* , pp. 331-337, 2007.

[19] David K. Y. Yau, John C. S. Lui, Feng Liang, and Yeung Yam, "Defending Against Distributed Denial-of-Service Attacks With Max-Min Fair Server-Centric Router Throttles," *IEEE/ACM TRANSACTIONS ON NETWORKING*, vol. 13, no. 1, pp. 29-42, Feb. 2005.
[20] Anukool Lakhina, Konstantina Papagiannaki, Mark Crovella, Christophe Diot, Eric D.Kolaczyk, and Nina Taft, "Structural Analysis of Network Traffic Flows," in *Proc. of SIGMETRICS/Performance*, New York, USA, pp. 61-72, 2004.
[21] Cisco IOS NetFlow White Papers, http://www.cisco.com/en/US/products/ps6601/prod_white_papers _list.html.
[22] Cisco NetFlow Performance Analysis White Papers, http://www.cisco.com/en/US/technologies/tk543/tk812/technologies_white_paper0900aecd802a0eb9_ps6601_Products_White_Paper.html, 2007.
[23] Simon Haykin, "Adaptive Filter Theory," Beijing: Publishing House of Electronics Industry, 2002.
[24] V. Paxson, "Bro: A System for Detecting Network Intruders in Real-time," *Computer Networks*, vol. 31, no. 23-24, pp. 2435-2463, 1999.
[25] Brett Ninness, Stuart Gibson, "The EM algorithm for Multivariable Dynamic System Estimation," *Technical Report EE200101*, 2001.
[26] R. H. Shmway, D. S. Stoffer, "Dynamic Linear Models with Switching," *Journal of the American Statistical Association*, vol. 86, no. 415, pp. 763-769, 1991.
[27] V. Digalakis, J. Rohlicek, M. Ostendorf, "ML Estimation of a Stochastic Linear System with the EM Algorithm and Its Application to Speech Recognition," *IEEE Trans. On Speech and Audio Processing*, vol. 1, no. 4, pp. 431-441, 1993.
[28] Douglas M. Hawkins, Peihua Qiu, Chang Wook Kang, "The changepoint model for statistical process control," *Journal of Quality Technology*, vol. 35, no. 4, pp. 355-366, 2003.
[29] D. Moore, G. M. Voelker, S. Savage, "Inferring internet Denial-of-Service activity," in *Proc. of the 10th USENIX Security Symposium*, pp. 9-22, 2001.
[30] Hao Jiang, Constantinos Dovrolis, "Why Is the Internet Traffic Bursty in Short Time Scales," in *Proc. of ACM SIG METRICS '05*, pp. 241-252, June 2005.
[31] J. Jung, B. Krishnamurthy and M. Rabinovich. "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites," in *Proc. of World Wide Web Conference*, Hawaii, USA, 2002.
[32] Anukool Lakhina, Mark Crovella, Christophe Diot, "Mining anomalies using traffic feature distributions," in *Proc. of SIGCOMM'05*, Philadelphia, Pennsylvania, USA, pp. 217-228, 2005.

Ruoyu Yan received a M.S. degree from Beijing Jiaotong University in computer science, China, in 2004. Currently he is a Ph.D. candidate in computer science, Xi'an Jiaotong University, China. His research interests focus on network security.

Qinghua Zheng received his B.S. and M.S. degrees in computer science and technology from Xi'an Jiaotong University, China, in 1990 and 1993, respectively, and his Ph.D. degree in systems engineering from the same university in 1997. He was a postdoctoral researcher at Harvard University in 2002. Since 1995 he has been with the Department of Computer Science and Engineering at Xi'an Jiaotong University, and was appointed director of the Department in 2008 and Cheung Kong Professor in 2009. His research interests include network security and intelligent e-learning.

Haifei Li is an associate professor of Computer Science at Union University, Jackson, TN, USA. He received his M.S. and Ph.D. degrees in Computer Science from the University of Florida, in 1998 and in 2001, received his Bachelor's degree in Computer Science from Xi'an Jiaotong University, Xi'an, China in 1990. Dr. Li's area of interest includes e-learning, database, e-commerce, automated business negotiation and business process management.