Early Detection of BGP Instabilities Resulting from Internet Worm Attacks

S. Deshpande
ECSE Department,
Rensselaer Polytechnic Institute
Troy, NY 12180

M. Thottan
Center for Networking Research,
Bell Laboratories
Holmdel, NJ 07733

B. Sikdar
ECSE Department,
Rensselaer Polytechnic Institute
Troy, NY 12180

***Abstract*— The increasing incidences of worm attacks in the Internet and the resulting instabilities in the global routing properties of the Border Gateway Protocol (BGP) routers pose a serious threat to the connectivity and the ability of the Internet to deliver data correctly. In this paper we propose a mechanism to detect/predict the onset of such instabilities which can then enable the timely execution of preventive strategies in order to minimize the damage caused by the worm. Our technique is based on online statistical methods relying on sequential change-point and persistence filter based detection algorithms. Our technique is validated using a year's worth of real traces collected from BGP routers in the Internet that we use to detect/predict the global routing instabilities corresponding to the Code Red II, Nimda and SQL Slammer worms.**

I. Introduction

With BGP being the default inter-domain routing protocol used in the Internet today, its stability and robustness is critical for ensuring the delivery of packets and maintaining connectivity. The dynamics of BGP also play an important role in the end user's perception of the network's performance since it controls the flow of inter-domain traffic. Recent analysis of BGP routing data [2] has shown strong correlation between instabilities in BGP routing tables and the propagation phases of well known worms like Code Red II and Nimda. Such large scale global routing instabilities result in widespread degradation in the end-to-end utility and significant problems at the links on the edges of the Internet. In the light of recent and increasingly frequent worm attacks, developing techniques for the early detection of impending instabilities caused by such attacks is crucial for initiating corrective action and thereby maintaining the health and operation of the Internet. This paper proposes a technique for on-line detection of BGP instabilities caused by worm attacks by using a statistical change-point detection technique.

Existing work on anomaly detection in computer networks has focused primarily on failure detection. The most common methods employed are rule-based approaches, finite state machine models, pattern matching and statistical analysis. Rule-based approaches are dependent on prior knowledge about fault conditions on the network and thus cannot capture the subtle nuances of evolving network environments [8]. Rule-based systems can be improved by using adaptive learning techniques [9]. However, this approach increases computation time and complexity. Anomaly or fault detection using finite state machines, models alarm sequences that occur during and prior to fault events [7]. Given a cluster of alarms the objective is to find the best explanation among them and no attempt is made to generate the individual alarms themselves which is a key component to any detection system. A pattern matching technique to detect network anomalies has been proposed and implemented in [10]. The efficiency of this approach depends on the accuracy of the traffic profile generated. In the face of evolving network topologies and traffic conditions, this method may not scale gracefully.

In this paper we propose online learning and statistical approaches using which, it is possible to continuously track the behavior of the network. We use a sequential change-point detection mechanism using non-parametric Cumulative Sum method based on auto-regressive modelling to detect Internet worm attacks and the instabilities they induce in BGP routers. We provide a characterization of the data obtained from BGP routers and highlight the distinguishing features that occur during a worm attack. We also present the detection algorithm and evaluate its effectiveness using real data traces obtained from BGP routers on the Internet.

The rest of the paper is organized as follows. Section II presents the relevant background information. Sections III and IV formally describe the problem and present the detection algorithm, respectively. Section V presents the results of our algorithm when applied to real BGP traces. Finally, Section VI presents the concluding remarks.

II. Background

Since the Morris worm attack of 1988, active worm attacks have made frequent appearances and in the recent past, have increased both in their frequency, spread and extent of damage caused. The most successful among these like the Code Red II, Nimda and the SQL Slammer worm employ a local scanning methodology to spread quickly and infect other machines. When such a worm is introduced in a network, it simultaneously scans many machines to find a vulnerable machine to infect. Once such a machine is successfully found and compromised, it executes a copy of the worm code and now starts scanning new machines to infect. The spreading rate of the virus depends on the scanning technique used, the rate at which security patches are applied and the population of vulnerable machines.

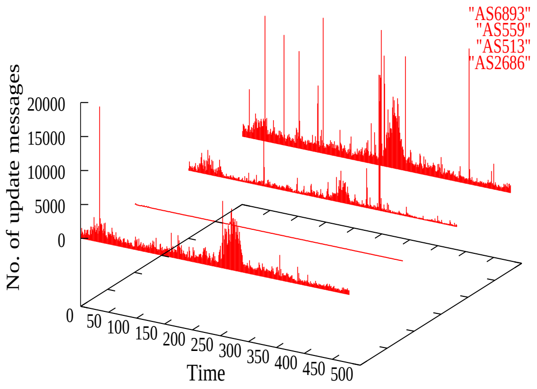


Fig. 1. BGP update message volumes.

BGP is the default inter-domain routing protocol in the Internet today and is based on the path vector routing mechanism [3].

While the overwhelming majority of the existing worms do not specifically target BGP routers, their impact on the network has been shown to create long term instabilities in BGP routers [2]. This is due to a number of direct and indirect causes that include: (1) congestion-induced failures of BGP sessions, (2) flow-diversity induced failures of BGP sessions due to router CPU overloads, (3) proactive disconnection of certain networks and (4) failures of other equipment at the Internet edge such as DSL routers. Also, some of the inherent features of BGP itself allow local connectivity dynamics to propagate globally. In order to prevent or minimize the damage caused by these worms on BGP routing, it is necessary to detect their onset as soon as possible. In the next sections, we develop a methodology for fast online detection of instabilities in BGP routers due to worm attacks.

III. Problem Description

Worm attacks on the Internet can lead to long term instabilities in global routing [2]. These instabilities are characterized by large and sustained exchanges of BGP routing updates, both route announcements and withdrawals, as shown in Figure 1. The figure shows the update messages received by a router in Geneva from four of its peers (in four different ASes) over a five day period: 17th-21st July 2001. The Code Red II worm attacked the Internet on the 19th July and we see the sharp and sustained increase in the message volume in this period (x-axis values between 250-285). We note that the update message volume can be rather noisy with random spikes. We also note that the plot corresponding to one of the ASes in Figure 1 does not show any variability at all. This is because this particular AS was continuously resetting its connection, probably due to a hardware problem. However, this highlights the need for a detection algorithm which is robust against such nuances of particular network specific conditions.

Close observation of the BGP data reveals that in general the increase in the number of announcements is more than that of the number of withdrawals. Furthermore, route withdrawals are typically followed by new announcements. Under worm attacks it is observed that the announcement and withdrawal messages experience long periods (in terms of hours) of sustained exponential growth. This growth phase was followed by an exponential decay in route changes. This distinguishing feature of the data in the presence of worm attacks can be used to design robust detection algorithms that are insensitive to changes inherent to the normal operation of the network.

In this work we propose that worm attacks can be modelled as sustained correlated changes in the number of announcements and withdrawal messages incident on a BGP router. The changes in the individual variables are modelled using an Auto-Regressive (AR) process [1]. The problem statement is as follows:

Given a sequence of BGP update messages sampled at a fixed interval, generate alarms that correspond to the onset of a worm attack.

IV. Algorithm

The model of sustained correlated changes described above is used to develop a detection scheme. The scheme is based on modelling the onset of instabilities in BGP routers due to worm attacks using correlated transient signals that are embedded in the BGP update messages. These correlations appear in the traces corresponding to the update message volumes across the peers connected to a router and the transients manifest themselves as abrupt and sustained as depicted in Figure 1. The temporal correlation between the update messages from different peers connecting to a router distinguishes the transients local to a peer and arising due to non-worm related causes from those intrinsic to worm attacks.

In this paper, we model the BGP update message volumes and the transients therein using auto-regressive processes of order p. The model of sustained correlated changes described above is used to develop a detection scheme following the procedure outlined in [13]. The scheme involves three steps which are described below:

Step(1): In the first step, statistical changes in the characteristics of the BGP update messages are captured using a hypothesis test based on the Generalized Likelihood Ratio (GLR) test [12]. Changes are detected by comparing the variance of the residuals obtained from two adjacent windows of data which are referred to as the learning ($L(t)$) and test ($S(t)$) window. Residuals are obtained by imposing an AR model on the time series data in each of the windows. Consider a learning window $L(t)$ and test window $S(t)$ of lengths N_L and N_S respectively. Then the learning window $L(t)$ can be shown as:

$$L(t) = \{l_1(t), l_2(t), \cdots, l_{N_L}(t)\} \tag{1}$$

Any $l_i(t)$ in the equation above can be expressed as $\tilde{l}_i(t)$ where $\tilde{l}_i(t) = l_i(t) - \mu$ where μ is the mean of the segment $L(t)$. Now, $\tilde{l}_i(t)$ is modeled as an AR process of order p with a residual error $\epsilon_i(t)$

$$\epsilon_i(t) = \sum_{k=0}^{p} \alpha_k \tilde{l}_i(t-k) \tag{2}$$

where $\mathcal{A}_L = \{\alpha_1, \alpha_2, \cdots, \alpha_p\}$ and $\alpha_0 = 1$ are the AR parameters. Assuming each residual time is drawn from an $\mathcal{N}(0, \sigma_L^2)$ distribution, the joint likelihood of the residual time series is given by

$$p(\epsilon_{p+1}, \cdots, \epsilon_{N_L}/\alpha_1, \cdots, \alpha_p) = \left(\frac{1}{\sqrt{2\pi\sigma_L^2}}\right)^{\acute{N}_L} e^{\left(\frac{-\acute{N}_L\hat{\sigma}_L^2}{2\sigma_L^2}\right)} \tag{3}$$

where σ_L^2 is the variance of the segment $L(t)$, $\acute{N}_L = N_L - p$ and $\hat{\sigma}_L^2$ is the covariance estimate of σ_L^2. Using a similar expression for the test window $S(t)$, the joint likelihood ν of the two segments $L(t)$ and $S(t)$ is given by

$$\nu = \left(\frac{1}{\sqrt{2\pi\sigma_L^2}}\right)^{\acute{N}_L} \left(\frac{1}{\sqrt{2\pi\sigma_S^2}}\right)^{\acute{N}_S} e^{\left(\frac{-\acute{N}_L\hat{\sigma}_L^2}{2\sigma_L^2}\right)} e^{\left(\frac{-\acute{N}_S\hat{\sigma}_S^2}{2\sigma_S^2}\right)} \tag{4}$$

where σ_S^2 is the variance of the segment $S(t)$, $\acute{N}_S = N_S - p$ and $\hat{\sigma}_S^2$ is the covariance estimate of σ_S^2. The expression for ν is a sufficient statistic and is used to perform a binary hypothesis test based on the Generalized Likelihood Ratio test. Under the hypothesis H_1 implying that a change is observed, we have $\mathcal{A}_L \neq \mathcal{A}_S$ and $\sigma_L^2 \neq \sigma_S^2$ implying that a change is observed between the two windows. In order to obtain a value for the generalized likelihood ratio η that is bounded between 0 and 1 and using the maximum likelihood estimates for the variance terms, the likelihood ratio is given by

$$\eta = \frac{\hat{\sigma}_L^{-\acute{N}_L}\hat{\sigma}_S^{-\acute{N}_S}}{\hat{\sigma}_L^{-\acute{N}_L}\hat{\sigma}_S^{-\acute{N}_S} + \hat{\sigma}_P^{-(\acute{N}_L + \acute{N}_S)}}. \tag{5}$$

where $\hat{\sigma}_P^2$ is the pooled variance of the learning and test windows. Using this approach, we obtain a measure of the likelihood of worm attacks for each of the peers. These indicators, which are functions of system time are updated every N_S lags and the worm attack indicators thus obtained from the individual peers are collected to form a change vector $\vec{\psi}(t)$. The change vector $\vec{\psi}(t)$ serves as a measure of the deviations observed in normal network behavior.

Step(2): The spatial dependence between the changes in the announcements and withdrawals obtained from different peers are incorporated using a co-relator matrix A. The matrix A is a $M \times M$ matrix where M is the number of distinct update message traces available from peers. A quadratic functional

$$f(\vec{\psi}(t)) = \vec{\psi}(t)A\vec{\psi}^{'}(t), \tag{6}$$

is used to generate a continuous scalar indicator of security breaches. A value of 0 for this functional represents a healthy network and a value of 1 represents a potentially compromised network.

Step(3): The final step in the algorithm is to capture the sustained changes in the BGP update messages. This is done with the help of a persistence filter. It can be expressed as:

$$t_a = inf\{t_i : \sum_i I(f(\vec{\psi}(t_i)) \geq \lambda) \geq L\} \tag{7}$$

where t_a is the earliest time at which the functional $f(\vec{\psi}(t))$ exceeds the threshold λ for a sustained period of time L. Each time the condition expressed in Equation (7) is satisfied we declare the network to be compromised. This also means that given an instance of an alarm condition if a second instance of an alarm condition occurs within a specified interval of $(\tau - 1)$ lags (say) then this indicates persistent abnormal behavior. Thus, by incorporating persistence the occurrence of false alarms can be significantly reduced.

The implementation of the change-point detection algorithm depends on the size of the learning window N_L, the size of the test window N_S and the order of the AR process p. A higher order of the AR process will model the data in the window more accurately but will require a large window size since a minimum number of samples are necessary to be able to estimate the AR parameters accurately. An increase in the window size will result in a delay in the predication of an impending fault. We experimentally optimize the two window sizes based on these constraints.

V. Experimental Results

Our detection algorithm was validated using year long traces of BGP update messages collected by the RIPE Network Coordination Center (NCC) [4]. We report on the performance of our algorithm for the Code Red II (July 2001), the Nimda (September 2001) and the SQL Slammer (January 2003) worm attacks.

A. Overview of the Traces

The traces used for validation of our algorithm were collected by the RIPE NCC. For the validation results presented in this paper, we have used the update data for the relevant periods from the data collected at site CIXP (CERN Internet eXchange Point), Geneva starting April 2001. In the traces from CIXP Geneva, routing updates were collected from seven peers, each corresponding to a different AS. The seven peers are AS513, AS559, AS2686, AS3303, AS6893, AS8327 and AS12350. The traces have a time-stamp accuracy of a second.

In this paper we use the aggregated number of update messages (including announcements and withdrawal messages) and the state change messages (session reset messages etc.) of all the peers in bins of 15 minutes. We have also extracted and analyzed traces per individual peer, again in bins of 15 minutes.

B. Validation Results

In Figure 2 we show the result of the change detection algorithm (step 1 of Section IV) for the Code Red II worm when applied to the withdrawal, announcement and the combined (total) messages exchanged. The data set used corresponds to the traces collected over the entire month of July 2001 where the attack occurred on 19th of July. The x-axis is marked in terms of the index of the 15 minute samples. The attack location (19th of July) corresponds to values of 1780-1850 that are marked by the circles. For the results shown, we used a learning window of 72, a test window of 36 and a persistence

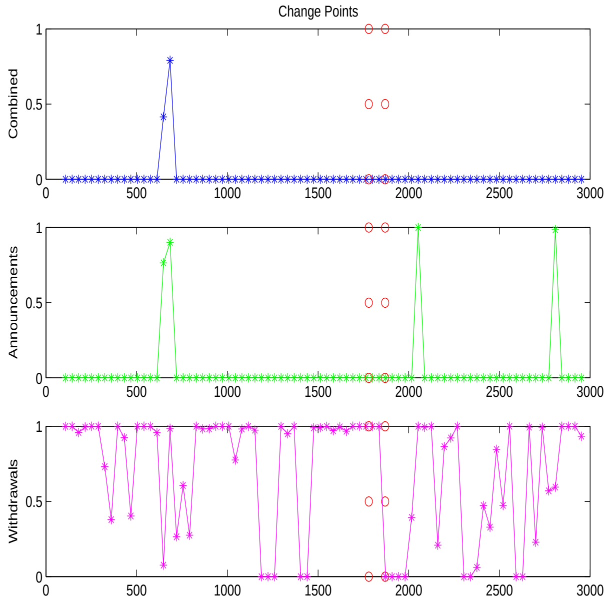


Fig. 2. Change point detection for BGP messages for the Code Red II worm (July 2001).

value of 4. We note that while the traces show change-points around the worm attack period, they also show many others which are not related to it. Also, we note that the combined trace shows change points at exactly the same instants as the announcement messages. This is because the volume of the announcements is much larger than those of the withdrawals and thus effectively masks the variations in the withdrawals. Since the combined trace adds no additional information as compared to the announcements, in the subsequent results, we only focus on the announcement and withdrawal messages.

The change-points not related to the worm attacks are eliminated from the detection mechanism using the second step of the algorithm in Section IV where we capture the correlation between the spatial dependencies of the withdrawal and announcement messages. The alarms obtained from comparing these correlations for the July trace is shown in Figure 3. Note that the co-relator amplifies only those changes that occur in both the announcements and the withdrawals. Finally, the result of incorporating the test for persistence in the changes (step 3 of the algorithm) is shown in Figure 4(a) where we plot the the final result of our fault detection algorithm (again, circles are used to demarcate the onset of the attack period). It is seen that our algorithm is able to detect and predict the onset of instabilities in BGP routing tables as a result of the worm attacks, thereby validating the proposed scheme.

In order to further validate the performance of our algorithm, in Figure 4(b) we show the predictive alarms generated by our algorithm for the Nimda worm and for the SQL Slammer worm in Figure 4(c). We used the month long trace collected for September 2001 for the Nimda worm (which took place during 18th-19th September)and for January 2003 for the SQL Slammer worm (the worm attack was on the 25th of January). In all these cases, we again see that our algorithm is able to correctly detect and predict the onset of

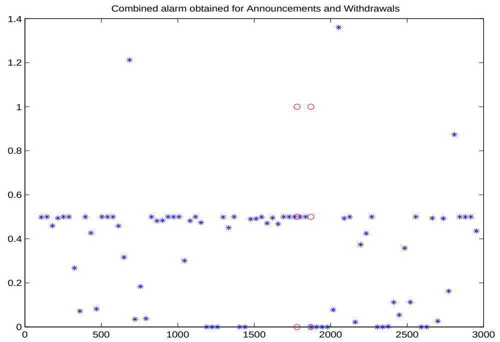


Fig. 3. Alarms obtained from correlating the change points in the withdrawal and announcement messages for the Code Red II worm. (July 2001)

instabilities due to the worm attack. However, we note that there are a couple of false alarms generated in the results for the SQL slammer worm. This was mainly due to the worm attack being detected sooner as compared to the older worms as a result of which the increase in the message volumes was not as sharp and sustained. This resulted in the selection of parameters which are prone to infer non-worm attack related transients in the traces as indications of worm attacks.

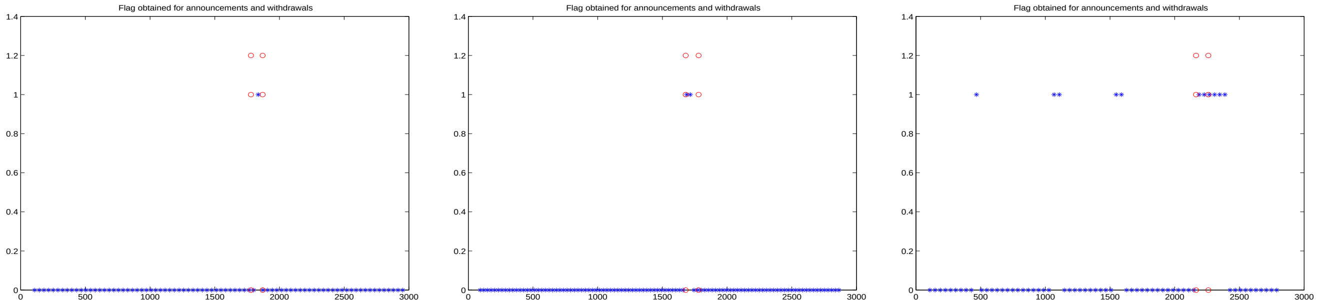
C. Results using Individual Peers

In this section, we present the results corresponding to the flags obtained when *only* the data from individual peers was considered instead of correlating them across peers. In Figure 5 we plot the flags generated from the update messages received from AS513 for the Code Red, the Nimda and the SQL Slammer worms respectively. Similarly, Figure 6 shows the corresponding results for AS559. The parameter settings for these results are given in Table I for the Code Red, Nimda and SQL Slammer worms, respectively.

From the results we can see that using just the information from individual ASes results in a large number of false alarms. These false alarms can be suppressed by correlating the flags from across peers as was done in the Figures 4(a), 4(b) and 4(c). Also, we can see that the parameter values for the three worms for a given AS lie in the same range. This suggests that the optimal parameter values vary from peer to peer and a system where different parameter settings are used for different peers would perform better than a system using a constant set of parameters across peers. We also note that different peers give better results for different worms indicating that some ASes might have been more affected by a particular worm than others. Also, we note that the value of the persistence filter depends on the worm. This is because different worms cause different levels of instabilities and thus to detect ones with low levels of activity, the persistence thresholds have to be lower.

VI. Discussion and Conclusions

In this paper, we proposed a mechanism to detect and predict instabilities arising in BGP routers as a result of worm attacks on the Internet. BGP being responsible for the inter-domain routing and determining the fate of a bulk of the data in the Internet coupled with the sharp increase in



(a) Code Red II (July 2001)

(b) Nimda (Sep 2001)

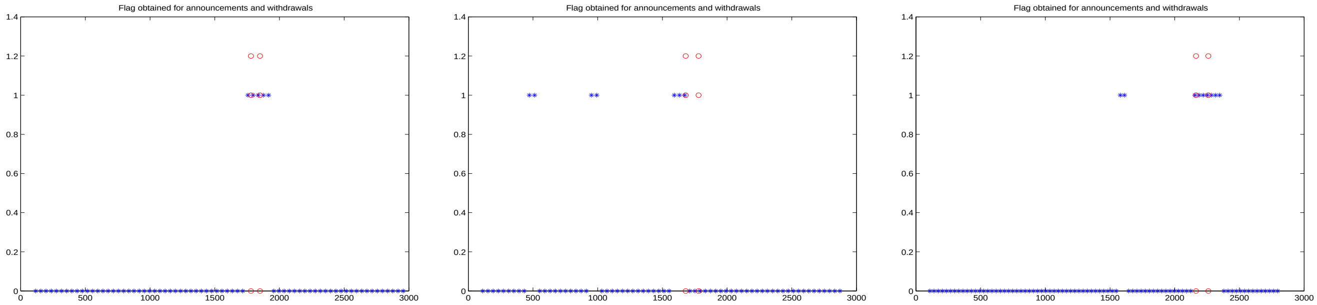
(c) SQL Slammer (Jan 2003)

Fig. 4. Flags corresponding to the different worm attacks across all the peers.

Value	Code Red II Worm Attack Location 1780-1850			Nimda Worm Attack Location 1680-1780			SQL Slammer Worm Attack Location 2165-2260		
	all-peers	AS513	AS559	all-peers	AS513	AS559	all-peers	AS513	AS559
Learning Window Size	72	76	76	64	72	72	68	76	76
Test Window Size	36	40	36	28	40	40	40	32	48
Persistence	4	6	5	11	9	8	9	12	8

TABLE I

THE PARAMETER SETTING VALUES FOR THE DIFFERENT DATA-SETS FOR THE DIFFERENT WORMS.(SEP 2001)

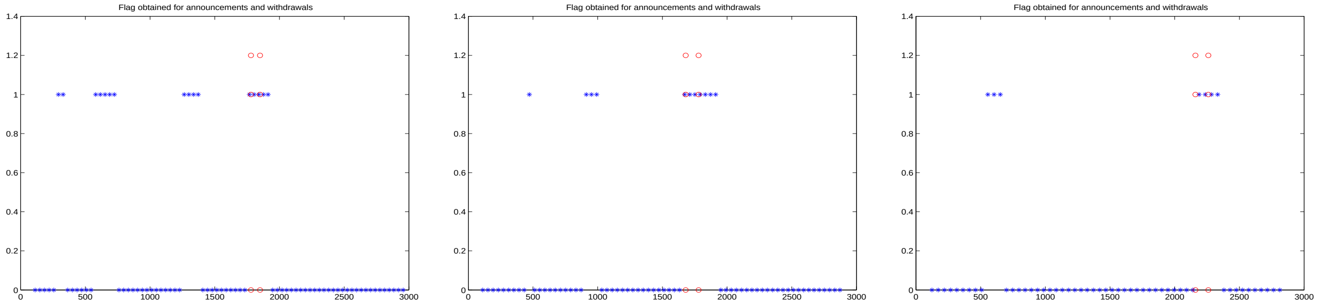


(a) Code Red Worm (July 2001)

(b) Nimda Worm (Sep 2001)

(c)SQL Slammer Worm (Jan 2003)

Fig. 5. Flags corresponding to peer AS513 for the different worms



(a) Code Red Worm (July 2001)

(b) Nimda Worm (Sep 2001)

(c)SQL Slammer Worm

Fig. 6. Flags corresponding to peer AS559 for the different worms

the number of worm attacks, makes detecting the onset of such instabilities at their earliest of critical importance for maintaining the connectivity and desired performance of the Internet. In this paper, we proposed a sequential change-point detection mechanism to detect and predict the onset of such instabilities. Our method uses the volume count of BGP update messages exchanged by the routers as an input. A test based on generalized likelihood ratios is used to generate change indicators for each peer. These indicators are then correlated across peers and passed through a persistence filter to generate alarm flags.

The proposed algorithm was tested using real data traces collected at various routers in the Internet over a one year period in 2001, 2002 and 2003. We showed that our technique is able to correctly detect and predict global routing instabilities from a number of Internet worms including the Code Red II, Nimda and SQL Slammer worms. Our results also showed the effectiveness of exploiting the correlations in changes in the BGP update message volumes across peers in the presence of worm attacks.

REFERENCES

[1] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes, Theory and Application,* Prentice Hall, 1993.

[2] J. Cowie, A. Ogielski, B. Premore and Y. Yuan, “Global routing instabilities during Code Red II and Nimda worm propagation,” Technical Report, Renesys Corporation, December 2001.

[3] Y. Rekhter and T. Li, “A border gateway protocol 4 (BGP-4),” RFC 1771, IETF, March 1995.

[4] Réseaux IP Européens Network Coordination Center, `http://www.ripe.net`.

[5] M. Thottan and C. Ji, “Anomaly detection in IP networks,” *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2191-2204, August 2003.

[6] L. Lewis, “A case based reasoning approach to the management of faults in communication networks,” *Proceedings of IEEE INFOCOM,* San Francisco, March 1993.

[7] I. Katzela and M. Schwarz, “Schemes for fault identification in communication networks,” *IEEE/ACM Trans. on Networking,* pp. 753-764, vol. 3, no. 5, October 1995.

[8] T. Ndousse and T. Okuda, “Computational intelligence for distributed fault management in networks using fuzzy cognitive maps,” *Proceedings of IEEE ICC,* pp. 1558-1562, Dallas, TX, June 1996.

[9] L. Lewis and G. Dreo, “Extending trouble ticket systems to fault diagnosis,” *IEEE Network*, vol. 7, no. 6, pp. 44-51, November 1993.

[10] F. Feather and R. Maxion, “Fault detection in an Ethernet network using anomaly signature matching,” *Proceedings of ACM SIGCOMM,* pp. 279-288, San Francisco, CA, September 1993.

[11] A. Franceschi, L. Kormann and C. Westphall, “Performance evaluation for proactive network management,” *Proc. of IEEE ICC,* pp. 22-26, Dallas, TX, June 1996.

[12] H. Van Trees, *Detection, estimation and modulation theory,* John Wiley and Sons, 1971.

[13] M. Thottan, “Fault detection and prediction for management of computer networks,” Ph.D Thesis, Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY, April 2000.