Real Time Detection of Link Failures in Inter Domain Routing

Xiaobo Long and Biplab Sikdar
Electrical, Computer and System Engineering
Rensselaer Polytechnic Institute, 110 8th Street, Troy NY 12180

***Abstract*— Measurements have shown that network path failures occur frequently in the Internet and physical link failures can cause network instability in large scale and severity. Inter domain routing protocols like Border Gateway Protocol (BGP) can take up to 15 minutes to converge after such failures and during the convergence period, packets may encounter transient loops, delays and losses [1]. Thus early anomaly detection mechanisms are of great importance. In this paper, we propose a Bayesian approach for time efficient link failure detection using BGP update message traces. The detection is done using an automated mechanism to label, train and classify the network status based on features extracted from BGP traces. In addition to detecting temporal changes in these features, our scheme augments its accuracy by including information on the spatial correlation of the route updates in the decision process. We validate our approach by testing the proposed mechanism on real BGP traces collected during three typical network outage events caused by link failures.**

I. Introduction

End users expect the Internet to provide a reliable packet delivery service. In practice, however, the Internet is a large scale loosely-coupled complex distributed system, made of many components which are not fully resilient against errors and failures. A key component gluing together this collection of domains and autonomous systems forming the Internet is the BGP routing protocol. With BGP being the default inter domain routing protocol in the Internet today, its stability and resilience to faults plays an important role in the end user's perception of network performance. Measurements show that various faults occur frequently almost everywhere in the Internet which affect BGP. Among these faults, physical link failures can cause large scale and severe network instability. Below are three cases of physical link failures in the Internet:

- Single link failure: During such events, BGP traffic is rerouted along alternate paths and may result in routing anomalies while BGP converges. If the single failed link happens to be the main cable connecting a big autonomous system or domain, such failures can be serious. For example, a submarine fiber cable in Pakistan was cut on June 27, 2005, which was Pakistan's main link with the Internet. The system took 12 days to repair and networks in Pakistan were unreachable from the rest of the world for two weeks.
- Simultaneous link failures: Usually physical failures happen independently in different geographical locations. However, in rare cases, failures do occur simultaneously. One such case was on June 21, 2005, when a rat chewed through one of the North Island, New Zealand's main communications cables while at the same time, a Telecom New Zealand workman accidentally damaged a second main cable in another part of the country. Telecom networks in New Zealand were paralyzed for five hours, knocking out mobile and Internet communications.
  Besides, power outages can cause large scale simultaneous failures of networking equipment. One case is when on May 25, 2005 in Moscow, Russia, the whole Moscow city was cut off electricity and Internet.
- Cascading failures: In a cascading network failure, a single failure (such as a fiber cut) results in widespread outage of the network. To the best of our knowledge, such collapses have not been seen in the Internet and general opinion is divided on the likelihood of their occurrence.

Link failures as above can result in network instabilities of various scale. They can cause routing loops in both link state and distance vector routing protocols. In distance vector routing protocols such as RIP, link failures may lead to counting-to-infinity looping. Unlike link state and distance vector routing protocols, BGP is a path vector protocol and a router sends an update message to its neighbors only when topology changes. When a destination becomes unreachable, BGP sends an explicit or implicit withdrawl message to its neighbors. BGP then converges only after the link failure information has been propagated over the entire network. Although a BGP node can eliminate routing loops, it suffers from slow convergence following a change in the network topology. Previous works have shown that network connectivity, failure location, and routing message queuing delay all effect BGP's convergence time [3] after link failures. During the convergence period, various anomalies may appear including abrupt increase in BGP traffic, routing oscillations, etc. and cause BGP instabilities.

To maintain the stability and efficiency of Internet in spite of link failures, network managers wish to detect failures as soon as they occur, as actions may then be required to resolve the problems. The goal of this paper is to provide a general technique for real time detection of the failures. How to improve BGP convergence performance during and after a failure is beyond the scope of this paper.

Since network outages are reflected in the departure of BGP update message traces (which include announcement and withdrawl messages) from their normal pattern in both volume and content, we detect abnormal behavior by observing targeted

BGP trace features and their correlations at monitoring nodes. We first extract features from BGP update traces to identify abrupt changes in time series BGP sequences. To complement the detection mechanisms for temporal changes in the BGP update messages, we incorporate information on the spatial correlations and patterns in the propagation of route updates in the decision process. These new features of the detection process are based on the fact that coherent departures from normal update patterns over a connectivity neighborhood (i.e. topology graph) are evidences of unusual events occurring in that neighborhood. Our work develops measures of the spatial correlation by treating measurements from each BGP peer as one feature and combining features from all peers. Alarms are then generated by classifying anomalies using pattern classification techniques. We validated the proposed mechanism by testing it on real BGP traces collected during three typical network outage events caused by physical link failures. Our detection results show that the approach achieves a low false alarm rate and a low miss rate when working on real test data.

The contributions of this paper are: **(i)** it provides a general approach to diagnose link failures from BGP message traces by separating data from a high-dimensional space into normal and abnormal subspaces, **(ii)** it provides tools for processing BGP traces that are fast enough to detect anomalies in real time and **(iii)** the proposed method is validated using real life data collected for three different link failure events.

The rest of the paper is organized as follows: Section II presents the related work. Section III describes the proposed methodology. In Section V we validate our approach by diagnosing three physical link outage events using real BGP data. Finally, Section VI presents the concluding remarks.

II. Related Work

Early work in anomaly detection was based on expert systems [13]. In expert systems an exhaustive database containing the rules of behavior of the faulty system is used to determine if an anomaly or fault occurred. Rule-based systems are too slow for real-time applications and are dependent on prior knowledge about anomalous conditions on the network. These rule-based systems rely heavily on the expertise of the network manager and do not adapt well to the evolving network environment. In some cases Fuzzy Cognitive Maps (FCM) are used to overcome this limitation [14]. Anomaly detection using finite state machines [15] model alarm sequences that occur during and prior to fault events. The difficulty encountered in using the finite state machine method is that not all anomalies or all instances of the same anomaly can be captured by a finite sequence of alarms of reasonable length. This may cause the number of states required to explode as a function of the number and complexity of the anomalies modeled.

A new approach proposed and implemented in [12] describes anomalies as deviations from normal behavior. In this approach online-learning is used to build a traffic profile for a given network. When newly acquired data fails to fit within some confidence interval of the developed profiles then an anomaly is declared. In the face of evolving network topologies and traffic conditions, this method may perform poorly and may not scale gracefully. A recent work on diagnosing traffic volume anomalies uses Principal Component Analysis (PCA) techniques [10]. The method is based on a separation of the high-dimensional space occupied by a set of network traffic measurements into disjoint subspaces corresponding to normal and anomalous network conditions by using Principal Component Analysis. However, PCA requires the use of SVD computation procedures which could become a bottleneck if applied to data with a large set of sources and destinations. In [11] the authors use a change point detection algorithm to detect several types of network anomalies. Here the sequential change detection is done using Auto Regressive (AR) models for the time series data. It shows that rigorous statistical analysis can lead to better characterization of evolving network behavior and eventually lead to more efficient methods for both failure and intrusion detection. Our work provides similar insight into the separation of high-dimensional data into normal and abnormal subspaces as [10]. But we use a simpler stochastic hypothesis test method for detection. We also provide tools for processing measurements that are fast enough to detect anomalies in real time.

III. Methodology

A. Feature extraction

The aim of our mechanism is to recognize periods of abnormal behavior corresponding to link failures using BGP data. As the first step, we need to extract features from the BGP traces which will facilitate the distinguishing of normal and anomalous periods.

To extract features that will be used in our detection mechanism, we first take a look at the nature of the messages exchanged by BGP peers during a period where a physical failure occurs. BGP messages exchanged during such periods involve withdrawl messages regarding the network prefixes that have lost connectivity and all the paths that have changed. In both cases, a router will receive announcement messages with increasing number of different ASPaths to the same set of prefixes (the prefixes of the affected networks) from all or most of its peers in a small interval of time. Figure 1 and Figure 2 respectively show the volume of BGP withdrawl messages and the number of different ASPaths for a given prefix in Pakistan during June 22, 2005 and July 11, 2005 from 32 BGP monitoring peers. The BGP update messages were collected by University of Oregon [8]. The link failure occurred on June 27 and its effect is reflected in the abrupt volume changes in both the figures in the period around June 27.

Based on the observations above, detecting patterns corresponding to messages with a large number of announcements of routes for a common set of prefixes from most of the peers with increasing number of ASPaths, followed by large number of withdrawl messages with routes that appear to be invalid, can thus be used as features to detect the presence of a physical failure. Also, all these messages will have the same AS number at the end of the ASPaths in the route, indicating within

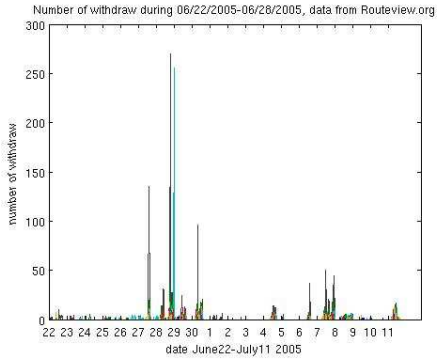


Fig. 1. Number of withdraw messages

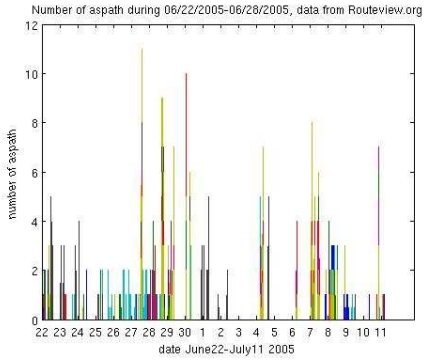


Fig. 2. Number of ASPaths

which AS the link failure occurred. Our selected features are described as follows:

1) The number of different ASPaths appearing in BGP Announcements for a targeted prefix during 15 minute time intervals.
2) The number of withdrawl messages appearing for a targeted prefix during 15 minute time intervals.

We now describe of proposed mechanism for feature extraction. In general, a BGP router has multiple peers and exchanges update messages with each of these peers. In our proposed methodology, we extract the above two features from each BGP monitoring peer. Let the number of peers of the BGP router be denoted by N. For each of the above two features, we obtain a feature vector, $\overline{x} = [x_1, x_2, \cdots, x_N]$, by combining the corresponding feature from all the peers, with each vector component representing a feature from one peer. In Algorithm 1 we present the pseudo-code for extracting the proposed features.

Although our selected features are effective in separating abnormal BGP data from normal ones, the features are usually corrupted by statistical noise, which may lead to errors in the detection process. To keep the fault detection process unaffected by this noise, we next develop a classifier which is insensitive to the noise. This classifier is then used to label the incoming data.

B. Data Labeling

We first train our classification model with a small set of typical data, which we refer to as the training data. Many pattern classifiers such as the Bayesian framework assume that the training samples used to design a classifier were labeled by their category membership. In practice, however, such information may not be available and in this paper we first apply an unsupervised learning procedure to label the training data using a clustering algorithm. Since in our case the data can have only two states, normal and abnormal, we group the incoming data into one of these two states by minimizing the sum of squares of distances between features of the training data (obtained using Algorithm 1 in the previous section) and the corresponding cluster centroid. This labeled data is then used as training data to train our classification model. The trained classifier is then used to detect link failures in the actual BGP data that we are interested in.

In a typical scenario, a BGP router may be connected to a number of peers and our scheme extracts features from the messages advertised by each of them. As a result, the dimensionality of the feature data used as input to the classifier, denoted by N in feature vector $\overline{x} = [x_1, x_2, \cdots, x_N]$, can be quite high. The classical Bayesian approach for hypothesis testing requires knowledge of the probability density function (PDF) of the data or knowledge of the sufficient statistic under class hypotheses. In the case of the relatively high-dimensional feature set we are dealing with, the performance of such an approach is severely limited by the ability to estimate the PDF on a high-dimensional space. The complexity of the high-dimensional space quickly overwhelms the ability to accurately estimate the distribution (also called the curse of dimensionality [16]). To circumvent this problem, before classification, we project the high-dimensional feature vector into a low-dimensional space. In the next section we show that

Algorithm 1 Feature Extraction Algorithm

N: `number of BGP peers`;
i: `index of ith peer`;
t: `time of new BGP messages arrival`;
x_i: `feature for peer i`;
f: `AS where the message is from, in 'ASxxx' format`;
p: `the prefix BGP message is announced for, in IP.IP.IP.IP/ format`;
P: `monitored target destination prefix, in IP.IP.IP.IP/ format`;
repeat
  obtain new collections of BGP messages from all peers every 15 minutes;
  convert binary BGP trace into txt file;
  for all i such that $0 \leq i \leq N$ **do**
    if p matches P **then**
      x_i++;
    end if
    store $x_i(t)$ in file;
  end for
until monitoring process terminated

this projection obtains low-dimensional sufficient statistics for class hypotheses.

C. Reducing Feature Dimensionality

In our proposed method for feature extraction, two feature vectors with dimension equal to the number of BGP peers is extracted at the monitoring point. To reduce the dimensionality of these feature vectors, we use a projection technique to transform them into a single dimension. The first step in our scheme is to obtain the projection direction $\overline{w}$ for transforming the high-dimensional feature data $\overline{x} = [x_1, x_2, \cdots, x_N]$ into lower dimensional data. As a BGP router continuously monitors the update messages from each of its peers, it generates a feature vector based on the observations of the last T time units ($T = 15$ minutes in our measurements). Applying Algorithm 1, we then obtain a time series of the feature vectors with $\overline{x_i}$ denoting the i-th feature vector and i denoting the discrete time index. During an observation time interval, say we obtain a data set $[\overline{x_1}, \overline{x_2}, \cdots, \overline{x_n}]$ of n samples which we use as the training data. This training data is then labeled using the technique described in the previous section. Let D_1 represent the subset of the training data which belongs to the abnormal class (i.e. contains a link failure event) and D_2 be the subset comprising the normal data. D_1 is labeled w_1 and D_2 is labeled w_2, where w_1 and w_2 represent the class of normal and abnormal data, respectively. If we form a linear combination of the components of $\overline{x}$, we obtain the scalar dot product

$$\overline{y} = \overline{w}^t \overline{x} \tag{1}$$

and a corresponding set of n samples $\overline{y_1}, \cdots, \overline{y_n}$ divided into the subsets Y_1 and Y_2, which belong to w_1 and w_2, respectively. Each $\overline{y_i}$ is the projection of the corresponding $\overline{x_i}$ onto a line in the direction of $\overline{w}$. The direction of $\overline{w}$ is selected to separate the projections of each classes as well as possible. It is proved in [9] that

$$\overline{w} = S_w^{-1}(\overline{\mu_1} - \overline{\mu_2}) \tag{2}$$

where $\overline{\mu_i}$ is the N-dimensional sample mean given by

$$\overline{\mu_i} = \frac{1}{n_i} \sum_{x \in D_i} \overline{x} \tag{3}$$

and

$$S_w = S_1 + S_2 \tag{4}$$

$$S_i = \sum_{x \in D_i} (\overline{x} - \overline{\mu_i})(\overline{x} - \overline{\mu_i})^t \quad i = 1, 2 \tag{5}$$

Using Eqn. (1) we can then project the high-dimensional features into one-dimensional data and use it as input to a Bayesian hypothesis test.

D. Hypothesis Test

In the last step of our scheme, we conduct a hypothesis test on the feature traces to detect the transition of the network from the normal to abnormal state. This test is based on the assumption that the one dimensional data obtained after the projection has a Gaussian distribution. Let the data corresponding to normal periods of the network's operation has pdf $p_0 = \eta_0(\mu_0, \Sigma_0)$, while during periods of abnormal behavior (i.e. link failure) the feature data has distribution of $p_1 = \eta_1(\mu_1, \Sigma_1)$. Since the data is Gaussian, its mean and variance (μ_0, Σ_0) and (μ_1, Σ_1) respectively, are sufficient statistics for two class hypothesis testing. The unknown change points from normal to abnormal operation of the network are estimated by comparing the posterior probabilities computed using Bayes theorem. The hypotheses to be tested are:

- $\mathcal{H}_0[\text{null}] : Y \sim \eta_0(\mu_0, \Sigma_0)$
- $\mathcal{H}_1[\text{alternative}] : Y \sim \eta_1(\mu_1, \Sigma_1)$

We assume that there exists an a priori probability associated with the hypothesis: $P(H_0) = \pi$ and $P(H_1) = 1 - \pi$. For simplicity, we assume that c_{ij}, the cost incurred by choosing hypothesis $\mathcal{H}_i$ when hypothesis $\mathcal{H}_j$ is true, has uniform cost. The likelihood ratio test between $\mathcal{H}_0$ and $\mathcal{H}_1$ is $L(y) = \frac{p_1(y)}{p_0(y)}$. Thus the corresponding Bayesian decision rule is:

$$\delta_B(y) = \begin{cases} 1, & \text{if } L(y) \geq \tau \\ 0, & \text{if } L(y) < \tau \end{cases} \tag{6}$$

For a given τ', the rule above can be proved to have a form as follows

$$\delta_B(y) = \begin{cases} 1, & \text{if } y \geq \tau' \\ 0, & \text{if } y < \tau' \end{cases} \tag{7}$$

The next step of our proposed method is to obtain the decision boundary τ' using the training data set. Once the decision boundary is obtained, for the new incoming test data, all we then need to do is to project the high-dimensional data onto low dimensional ones with $\overline{w}$, then compare its value to the decision boundary to make a decision.

Let $g_i(\overline{y})$, $i = 1, 2, \cdots$ denote the discriminant function for data in class i. Then we have

$$g_i(\overline{y}) = -\frac{1}{2}(\overline{y} - \overline{\mu_i})^t \Sigma_i^{-1}(\overline{y} - \overline{\mu_i}) - \frac{1}{2} \ln |\Sigma_i| + \ln P(w_i)$$

For every observation y, Bayes decision rule is to assign a data y to class w_i if

$$g_i(y) > g_j(y) \quad \forall \quad j \neq i \tag{8}$$

We can now calculate the Bayesian decision boundary y_0 using $g_i(y)$ and y_0 is the decision boundary τ'. In Algorithm 2 we present the overall pseudo-code for detection the link failure and alarm generation.

E. Computational Complexity

The computational complexity of the proposed mechanism is of considerable practical importance in order to evaluate the feasibility of its deployment. In general, we care about how the algorithm scales as the feature dimension increases.

Algorithm 2 Process for Anomaly Detection and Alarm Raise

$\overline{x}_{N\times 1}$: N-dimensional feature vector;
$\overline{X}_{N\times n}$: feature vector sequence of n samples;
$\overline{w}$: projection direction;
$Y_{1\times n}$: feature vector sequence after projection;
τ': Bayes decision boundary;
$A(k)$: alarm from feature k;
A: final alarm indicating a link failure;
for all features **do**
  DATA TRAINING
  use clustering to classify n samples into normal and abnormal classes;
  obtain $\overline{w}$;
  for all i such that $0 \leq i \leq n$ **do**
    $Y(i) = \overline{w}^t \overline{X}(i)$;
  end for
  use Bayes discriminant function test to obtain τ';
  DATA TESTING
  get new test data $\overline{x}$;
  $y = \overline{w}^t \overline{x}$;
  if $y > \tau'$ **then**
    $A(k) = 1$;
  else
    $A(k) = 0$;
  end if
end for
$A = 1$;
for all feature $A(k)$ **do**
  A = A AND A(k);
end for

Moreover, we are typically less concerned with the complexity of learning process which only needs to be done only, at the beginning of the detection process. We are more concerned with the complexity of making a decision, which needs to be done online as new update messages and the corresponding measurements are received.

In our mechanism, we used a clustering algorithm to classify and label the training data. Although this algorithm is computationally intensive, it needs to be done only once. Then we converted the high dimensional feature vectors into a single dimension. In the next step, the decision boundary τ' under a Bayesian framework was obtained using the training data set. For the new incoming testing data, all we then need to do is to project the high-dimensional data onto one-dimensional ones with $\overline{w}$, then compare its value to τ' to make a decision, which takes little computational effort. If the incoming data is classified as abnormal, an alarm is raised. This computational ease of our mechanism allows it to work efficiently and makes it suitable for the online detection of link failures using BGP data.

IV. Generalization of Detection Method

A BGP router typically deals with more than one hundred thousand different address prefixes. These prefixes need to be simultaneously treated by the detection mechanism and all these treatments should be used together to detect and identify a given link failure. One could detect link failures by collecting BGP message traces for all prefixes, and applying temporal statistical detection methods to each prefix. In general, this is impractical given the large number of prefixes to deal with since monitoring all prefixes is extremely resource intensive. Instead, we develop a simpler and more practical technique for diagnosing link failures. Given that a volume anomaly propagates through the network, we make use of the fact that we should be able to observe the anomaly on most prefixes located in the same geographic area or same AS number when a link outage occurs. What's more, the same parameters in the Bayesian hypothesis test can be used for all prefixes in the same geographic area within the same AS. These parameters need to be updated to reflect current network status for each new link failure event detection. To complete the detection algorithm described in the previous section, our generalized algorithm 3 is given in Algorithm 3.

Algorithm 3 Generalized Anomaly Detection Process

repeat
  Monitor the count of different ASPaths on BGP routers;
  if abrupt volume change detected **then**
    Extract target problem AS number from ASPath;
    for targeted AS(i) **do**
      choose a small subset of prefix in AS(i);
      for all prefix in subset **do**
        use BGP trace for the target prefix as both training and testing data;
        if alarms raised **then**
          update parameters in Bayesian hypothesis testing;
          break;
        end if
      end for
      for all the remaining prefixes in AS(i) **do**
        apply Bayesian hypothesis test to detect anomaly;
      end for
    end for
  end if
until monitoring process terminated

V. Validation

Three network outage events that occurred over the past year are used to test our detection algorithm. The first event is that of a submarine fiber cable cut in Pakistan on June 27, 2005. The cable was Pakistan's main link to the Internet and the system took 12 days to repair and was finally fixed on July 8, 2005. During this period, networks in Pakistan were unreachable from the rest of the world. The second

outage incident on which we tested our mechanism occurred on May 25, 2005 in Moscow, Russia. The whole Moscow city was cut off electricity and Internet on that day. Parts of the Moscow region and the Tula region were also affected. The third event that we consider is that of June 21, 2005, when a rat chewed through one of the North Island, New Zealand's main communications cables while at the same time, a Telecom New Zealand workman accidentally damaged a second main cable in another part of the country. The resulting outage lasted for five hours and affected both mobile and wired communications. The BGP updates message that were used as inputs to our detection mechanism were obtained from `www.routeviews.org`, collected by the University of Oregon. The BGP messages were collected from 32 BGP peers, which involve several locations around the Internet. Note that while the data traces available from the University of Oregon are updated after noticeable delays and might not capture the present picture of a network but reflect a past image, our proposed method is designed to run in actual BGP routers which obtain update messages from its peers in real time. Thus in actual deployments of our scheme, the update message traces would be available at the routers without any delay, facilitating the real time detection of link failures.

A. Training Data

The training data that we used for initially labeling the data using the clustering algorithm described in Section III-B and then for obtaining the projection direction $\overline{w}$ and the Bayesian decision boundary τ' corresponds to 21 days of BGP traces for the period between June 22, 2005 and July 11, 2005. This period includes the outage in Pakistan and the training data set we use is BGP update messages with destination prefix 202.163.120.0/24, which is the Cyber Internet Services (Pvt) Ltd. network in Karachi, Pakistan. The traces from each of the 32 BGP monitoring peers was processed every 15 minutes to obtain the time series of the feature vectors. We first extracted the feature corresponding to the number of different ASPaths for the given prefix, from each peer and considered it as one component of the feature vector $\overline{x}$. Thus $\overline{x}$ is a 1×32 dimensional data. We then obtained a time series of the feature vector $\overline{x_1}, \cdots, \overline{x_n}$ for the 21 days observation time period to obtain the projection direction $\overline{w}$ using the methodology outlined in Section III-C. After assuming that the apriori probabilities are $P(w_1) = P(w_2) = 1/2$, the Bayesian decision boundary τ' was then calculated by comparing $g_1(y)$ and $g_2(y)$. This τ' as a linear boundary divides the sample space into two subspaces: normal and abnormal, and will be used as decision boundary for the test data. This procedure was then repeated for the feature vector corresponding to the number of withdrawl messages for the given prefix.

B. Test Results

We now present the results obtained after applying our mechanism to detect the three link failure incidents mentioned previously, starting first with the Pakistan event. For testing the performance of the proposed mechanism, we used a different network prefix, namely 202.165.232.0/24, which is Apollo Telecom (Pvt.) Ltd. in Islamabad, Pakistan. The detection results are shown in Figure 3. The area between the two vertical lines shows the period during which the link was down. Figure 3 (a) shows the alarms generated when feature vectors corresponding to the ASPath count is used for detection. As can be seen, a number of alarms are generated within the anomaly period with zero false alarms, validating the proposed detection mechanism.

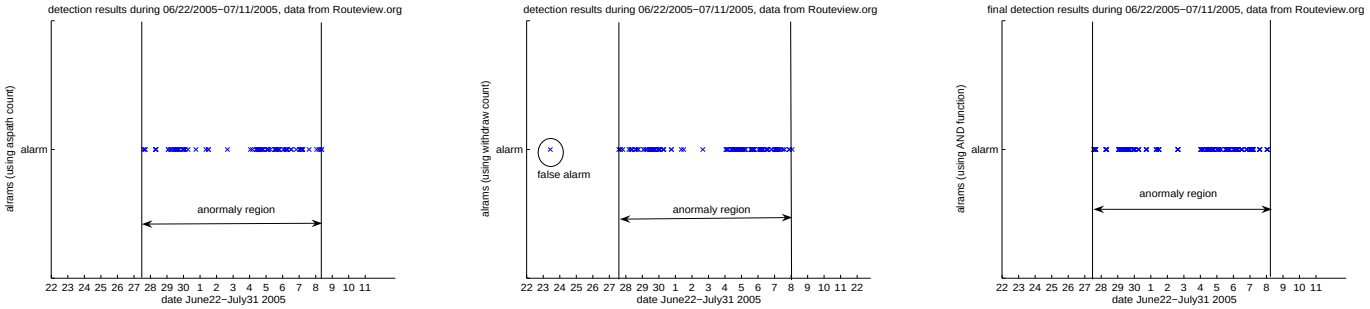
Detection result for the case where the number of withdrawl messages from each peer is used as the feature vector is shown in Figure 3 (b). We observe that while we still obtain alarms in the period when the link was down, there is one false alarm. The probability of such errors can be reduced and the detection mechanism can be made more robust if the detection results from the two features are combined to give the final alarm. This is achieved by clustering the alarms from the two features. For each instant when the detection process is invoked, we now also consider the past results of the detection process in an interval of length τ. If there is any alarm raised in this interval τ, we set $D_i(\tau) = 1$, or else we set $D_i(\tau) = 0$ where $i \in \{0, 1\}$ represents which feature was used for the detection process. The final alarm is generated if and only if detection using both the features gives an alarm i.e., the final detection rule is $D(\tau) = D_0(\tau)$ AND $D_1(\tau)$ where D_0 represents detection using ASPath count and D_1 is detection using withdrawl count. This clustering mechanism achieves accurate detection results and a low false alarm rate. Figure 3 (c) shows the results obtained using our clustering mechanism with $\tau = 4$. We note that our final detection mechanism accurately detected the anomalies during June 27, 2005 and July 8, 2005, without any false alarms.

To further analyze the performance of our algorithm, we also applied it on two other outage events: May 25, 2005 Moscow outage and June 21, 2005 New Zealand fiber cut. For the Moscow outage, the prefix used for training data is 194.220.204.0/23 located in Moscow, Russia and the Testing data is for prefix 194.85.113.0/24, which is in Khabarovsk, Russia. Figure 4 (a), (b) and (c) show detection results when the ASPath count, the withdrawl message count and the clustering of the alarms from the two traces, respectively, are used for detection. Again we observe that the false alarm arising while using the withdrawl message count feature is eliminated by using the clustering mechanism to generate the final alarm.

For the June 21, 2005 New Zealand outage, our training prefix is 210.54.10.0/23 while the testing prefix is 210.54.134.0/24, both of Telecom New Zealand Ltd, Auckland, New Zealand, with detection result shown in figure 5 (a), (b) and (c) using the ASPath count, withdrawl count and the final detection result using the clustering mechanism. Again we note that the final result has no false alarms.

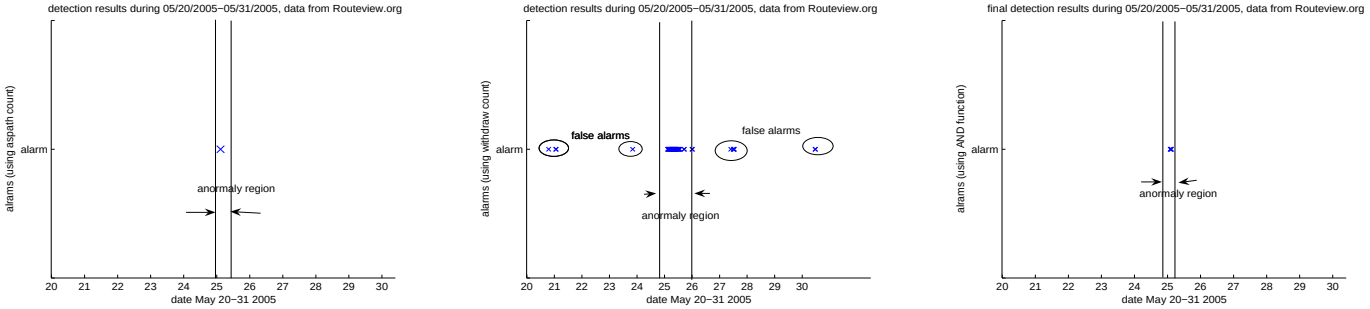
C. Probability of Error

In the Bayesian framework used in this paper, there are two ways in which a classification error can occur: either an



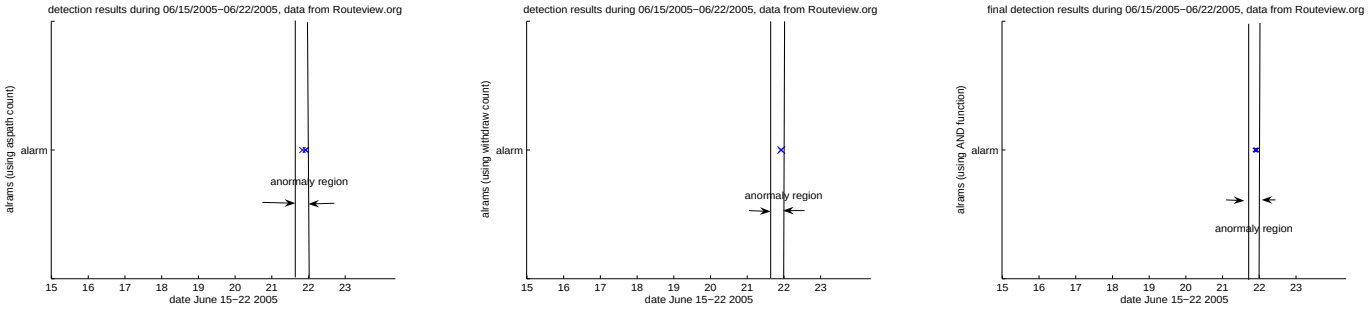
(a) Alarms using ASPath count (b) Alarms using withdrawl count (c) Final detection results

Fig. 3. Detection results for the Pakistan cable cut.



(a) Alarms using ASPath count (b) Alarms using withdrawl count (c) Final detection results

Fig. 4. Detection results for the Moscow outage.



(a) Alarms using ASPath count (b) Alarms using withdrawl count (c) Final detection results

Fig. 5. Detection results for the New Zealand outage.

observation y falls in w_2's region and the true state of nature is w_1, or y falls in w_1 and the true state is w_2. Since these events are mutually exclusive, the probability of error is

$$\begin{aligned} P[e] &= P(y \in w_2, w_1) + P(y \in w_1, w_2) \\ &= P(y \in w_2|w_1)P(w_1) + P(y \in w_1|w_2)P(w_2) \\ &= \int_{y \in w_2} p(x|w_1)p(w_1)dy + \int_{y \in w_1} p(x|w_2)p(w_2)dy \end{aligned} \quad (9)$$

Bayesian optimal decision boundary gives the lowest probability of error by choosing the decision boundary which make the $P[e]$ un-reducible. If we assume $P(y|w_1)$ and $P(y|w_2)$ to be Gaussian, the probability of error can then be calculated using the distribution function of Gaussian variables. However, since this is difficult to calculate, we use the Bhattacharyya Bound to estimate the error probability:

$$\begin{aligned} k_{half} = &\frac{1}{8}(\mu_2 - \mu_1)' \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\mu_2 - \mu_1) \\ &+ \frac{1}{2} log \left(\left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} / \sqrt{|\Sigma_1||\Sigma_2|} \right) \end{aligned} \quad (10)$$

and

$$P_{error} = \frac{1}{2} exp(-k_{half}) \quad (11)$$

In our detection for the Pakistan outage using number of ASPaths as features, we obtained the sample means μ_1 = -1.3658 and μ_2 = -354.3504; and sample covariance Σ_1 = 327.8203 and Σ_2 = 25.1643. Using Eqns. (9) and (10) the theoretical probability of error is 1.6978×10^{-39}, which is rather small. We also calculated the probability of error

by comparing K-Means classification results with Bayesian classification results. $P_{error} = 0.032$ in our tests for the Pakistan event. Probability of errors in other two tests for Moscow and New Zealand outage is also small, which are 0.001 and 0.0012, respectively, using number of ASPaths as features.

VI. Conclusion and Future Work

In this paper we presented a pattern classification based mechanism for detecting network link failures using BGP traces. Our data set covers more than 40 days of data, collected from 32 BGP monitoring nodes from various locations in the world. Three typical network link failure events are used to test our detection algorithm. Our detection results show that the proposed mechanism is very effective and has a low error probability and low computational complexity.

References

[1] D. Pei, X. Zhao, D Massey and L. Zhang, "A study of BGP path vector route looping behavior," *Proceedings of IEEE ICDCS*, pp. 720-729, Tokyo Japan, March 2004.

[2] N. Feamster, D. Andersen, H. Balakrishnan, and M. Kaashoek, "Measuring the effects of Internet path faults on reactive routing," *Proceedings of ACM SIGMETRICS*, pp. 126-137, San Diego, CA, June 2003.

[3] D. Pei, B. Zhang, D. Massey and L. Zhang, "An analysis of convergence delay in path vector routing protocols," *Computer Networks*, pp. 398-421, vol. 50, no. 3, February 2006.

[4] M. Thottan and C. Ji, "Anomaly detection in IP networks," *IEEE Trans. on Signal Processing*, pp. 2191-2203, vol. 51, no. 8, August 2003.

[5] V. Alaron-Aquino and J. Abarria, "Anomaly detection in communication networks using wavelets," *IEE Proceeding on Communications*, pp. 355-362, vol.148, no.6, December 2001.

[6] L. Ho, D. Cavuto, S. Papavassiliou and A. Zawadzki, "Adaptive and automated detection of service anomalies in transaction-oriented WAN's: Network Analysis, Algorithms, Implementation, and Deployment," *IEEE Transactions in Communications*, pp. 744-757, vol. 18, no. 5, May 2000.

[7] K. Zhang, A. Yen, X. Zhao, D. Massey, F. Wu and L. Zhang, "On detection of anomalous routing dynamics in BGP," *Proceedings of IFIP Networking*, pp. 259-270, 2004.

[8] `http://archive.routeviews.org/bgpdata/`

[9] R. Duda, P. Hart and D. Stork, *Pattern Classification*, Second Edition, Wiley-Interscinece Publication, 2001.

[10] A. Lakhina, M. Crovella and C. Diot, "Diagnosing network-wide traffi c anomalies," *Proceedings of ACM SIGCOMM*, Portland, OR, August 2004.

[11] M. Thottan, *Anomaly detection and prediction for management of computer networks*, Ph.D Thesis, Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY, April 2000.

[12] 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.

[13] L. Lewis, "A case based reasoning approach to the management of faults in communication networks," *Proceedings of IEEE INFOCOM*, pp. 1422-1429, San Francisco, CA, March 1993.

[14] 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.

[15] I. Katzela and M. Schwarz, "Schemes for fault identifi cation in communication networks," *IEEE/ACM Transactions on Networking*, pp. 753-764, vol. 3, no. 5, October 1995.

[16] Baggenstoss, P.M, "Class-specifi c feature sets in classifi cation," *Proceedings of IEEE ISIC*, pp. 413-416 , Gaithersburg, Maryland, September 1998.