A Study of Internet Routing Stability Using Link Weight

Mohit Lad [*] Jong Han Park [*] Tiziana Refice [†] Lixia Zhang [*]

ABSTRACT

The global Internet routing infrastructure is a large scale distributed system where routing changes occur all the time. While prior work on Internet routing dynamics examined routing stability to individual destinations, in this paper we study the routing stability of the Internet as a whole. We use the observed changes in the number of routes over each AS-AS link as a metric and measure such changes from multiple vantage points over a period of one year. We then apply Principal Component Analysis to identify those AS links that were most involved in routing changes. Our work is the first to combine measurement data collected from multiple monitors to gauge the overall routing stability in the Internet. Our results show that very few routing events impact the entire Internet, and those events were due to announcement of new prefixes either in the form of route leakages or address space de-aggregation. We also find that the impact of most routing events is confined to a small scope, and the existence of unstable AS links over long periods of time. We believe our approach represents a new direction in routing stability measurement and our findings shed new insight into the routing system performance.

1. INTRODUCTION

The Internet routing infrastructure is composed of a large number of independently administrated networks called Autonomous Systems or ASes, with Border Gateway Protocol (BGP) as the global routing protocol. In a system as big, complex and distributed as the Internet, routing changes are known to happen all the time. Routing instabilities in the Internet can affect data delivery often resulting in longer latencies or even dropped packets, thus degrading end to end performance. While prior studies have focused on the routing stability of individual destinations [20], the dynamics of individual prefixes are often not representative of what is going on in the Internet. To the best of our knowledge, no study has been done on the routing stability of the Internet as a whole. The difficulty to capture routing dynamics across multiple prefixes using a single metric, as well as the need to combine such information from multiple vantage points presents challenges in carrying out such a study. A study of routing stability would provide a good understanding of the limitations of current Internet routing as well as help in the design of future routing protocols.

There are many challenges one faces in conducting such a study of routing stability. First is the sheer number of routes that exist in the global routing table. A BGP router may contain routes to as many as 250,000 BGP prefixes, with tens of thousands of unique routes to these destinations. Thus, examining paths to individual prefixes simply does not scale. Further, routes are constantly changing resulting in tens of thousands of BGP updates processed by each BGP router every day, and this high volume of route changes over time presents another challenge. Most importantly different parts of the Internet see very different routing changes. Some events may only affect a small portion of the Internet, while others may cause a disturbance that is more globally felt. To tell which is which, requires painting a complete picture by combining routing changes observed at different portions of the Internet.

In this paper, we propose a new methodology for understanding routing instabilities affecting multiple prefixes. To scale with prefixes, we first need a metric, and our metric is motivated by the concept of *link weight* proposed in [9, 10], that represents the number of routes carried over an AS-AS link as computed from the routing table of a BGP router. By measuring the changes in these number of routes on each link seen by this BGP router, we can naturally capture aggregate behavior across multiple prefixes and provide a quantitative metric for routing stability. Ideally by looking at routing changes over each link, we would like to identify the links that stand out in terms of routing changes from the rest. One can imagine each monitor as a dimension, and examining changes along the individual vantage points can be cumbersome and may not produce desired result. For example, a link that stands out

[*] University of California, Los Angeles, USA. Email: {mohit,jpark,lixia}@cs.ucla.edu

[†] Roma Tre University, Rome, Italy. Email: refice@dia.uniroma3.it

in terms of routing changes from one monitor may be very different from a link that stands out when looking at multiple vantage points together. Our next big challenge is to be able to collectively examine the routing changes observed by BGP routers in different parts of the Internet. For this purpose, we use a statistical technique called Principal Component Analysis (PCA) to transform the original dimension to a new dimension space where individual axes are the combinations of different vantage points, thus making the examination of the routing changes more meaningful in terms of commonality as well as diversity observed by different vantage points.

We apply our scheme on BGP data collected from different vantage points over a period of one year from January 2007 to December 2007. Our goal is to find how often large scale routing instabilities occur and what is the scope or spread of these instabilities. We found quite a few cases of large scale routing problems that impacted a significant portion of the Internet. We found that the most widespread impact is caused when routing changes involve a link close to the origin AS announcing lots of prefixes. In particular, we found that the most widespread impacts were caused by routes being de-aggregated or internal AS routes being leaked out to the rest of the Internet. We also observed quite a few instances of the same routing instability repeating many times over our observation period. While a majority of them are restricted in their scope, a few impact a significant portion of the Internet. Overall, we find that Internet routing is pretty stable in terms of globally routing instabilities, but there are lots of instabilities that affect different small portions of the Internet. We would like to emphasize that in this paper we do not attempt to perform root cause analysis of routing instabilities. Rather, we focus on understanding *how routes shift* due to any routing events. In some sense, one can think of this as understanding the *effect* of routing events on the Internet, rather than what is the *root cause* or *origin* of the observed updates.

The rest of the paper is organized as follows. In Section 2, we briefly describe the basics in BGP routing and the metric of link weight. Section 3 describes how we apply PCA on our data set, and explains how we pick outlier links and interpret the results. In Section 4, we provide a basic validation of our scheme using simulation experiments under controlled settings. In Section 5, we describe preparation of the BGP data set spanning one year, in particular how we pick a good set of vantage points. In Section 6, we present our results on routing stability. Section 7 presents some open issues and limitations of our results. Finally, Section 8 presents related work, and Section 9 concludes the paper.

2. CAPTURING INTERNET ROUTING CHANGES

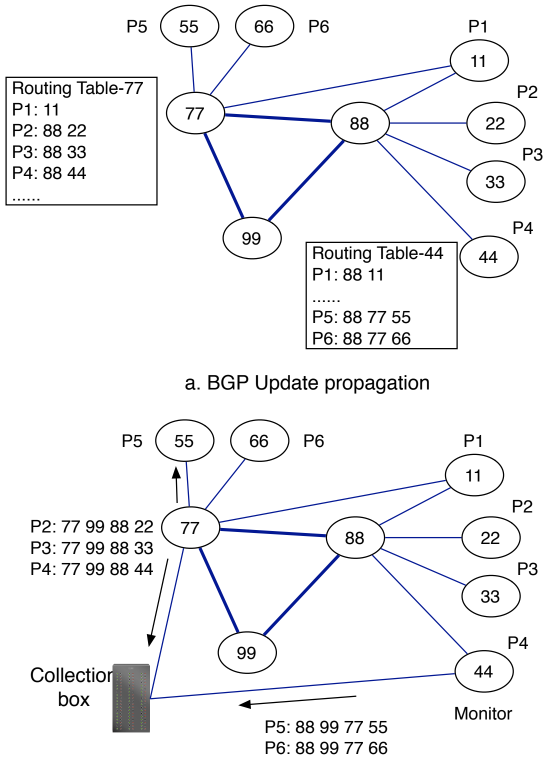
The Internet consists of a large number of administrative domains called autonomous systems (AS). Each AS is identified by an AS number and contains one or multiple destination networks. Each destination network is represented by an IP address prefix (e.g. M.I.T. announces 18.0.0.0/8). As of Dec 2007, the Internet routing system includes over 28,000 autonomous systems and over 250,000 prefixes.

Border Gateway Protocol (*BGP*) [19] is the de-facto routing protocol used between ASes to exchange information about prefixes reachability. Whenever a new prefix is announced in the Internet, BGP update messages are used to propagate routes to this prefix. For example in Figure 1a, AS 11 announces prefix P1 to its upstream (service provider) AS 88 and AS 77, who in turn prepend its own AS number to the path and propagates this path to its own neighbors. Route selection and propagation in BGP are determined by networks' routing policies, where the business relationship between two connected ASes plays a major role. AS relationship can be generally classified as customer-provider or peer-peer. In a customer-provider relationship, the customer AS pays the provider AS for routing traffic. The peer-peer relationship does not usually involve monetary flow; The two peer ASes exchange traffic between their respective customers only. In Figure 1, AS 11 is a customer of AS 77, and hence even though AS 77 receives a peer route via AS 88, it prefers to use the customer route to AS 11. The routing table at AS 77 in Figure 1a shows that AS 77 reaches prefix P2 using AS 88 as the next hop.

Note, two autonomous systems may connect at multiple physical locations through different BGP routers. For simplicity we refer to the routing table of a particular router in an AS as the routing table of that AS, and abstract the parallel connections between two ASes as a single logical connection.

2.1 Routing Instabilities

Whenever a BGP router's route to any destination prefix changes, it sends a BGP update to its neighbours. Routing changes can be caused by various reasons, such as link failures, BGP session resets[23], or policy changes. In Figure 1a, assume the BGP peering between 77 and 88 fails. As a result, AS 77 cannot use AS 88 as next hop to reach a bunch of prefixes, i.e. $P_2 \ldots P_4$. AS 77 then switches to using AS 99 to reach these prefixes and sends BGP update messages to AS 55, AS 66 and AS 11 communicating the new route to reach prefixes $P_1..P_4$ as shown in Figure 1b. Similarly, AS 88 will also send updates to its neighbors since its route to prefixes P5 and P6 will not be valid after failure of 77-88.



a. BGP Update propagation

b. Route changes due to link problems

Figure 1: Internet routing and BGP monitoring

Public data collection projects like RouteViews [18] and RIPE NCC's *Routing Information Service (RIS)* [17] connect to BGP routers in different autonomous systems and passively collect BGP updates from them. These BGP routers serve as vantage points and in the rest of this paper we refer to them as *monitors*. In Figure 1b, AS 77, AS 99 and AS 44 are connected to a collection box and receive BGP updates for this particular event. Note, that in this case, AS 88 sends updates for a different set of prefixes than that sent by AS 77. On the other hand AS 99 does not use the link 77-88 and though it receives updates from AS 77 and AS 88, it does not send any updates to its neighbors. Clearly, even for the same event, different ASes may see a different effect. From Figure 1b, we can see that while monitors inside AS 77 and AS 44 sent BGP updates to the Collection box following failure of link 77-88, AS 99 did not send any updates for the same event. Next, we discuss the metric which helps us capture the shifting of routes over AS links.

2.2 Link Weight Changes

As seen in Figure 1, when multiple routes are affected, it is difficult to aggregate them by their prefix identifiers. However, usually the affected routes share common AS links, and we take advantage of this fact to aggregate routing changes. We use the measure of link weight similar to that proposed in [9, 10] where each link is weighed by the number of AS routes carried as seen by an observation point. In this case, initially link 77-88 carried 3 routes as seen by AS 77, and hence had $wt(77, 88) = 3$. Similarly, $wt(77, 99) = wt(99, 88) = 0$ as seen by AS 77. When the link 77-88 breaks down, routes use AS 99 as the next hop to reach AS 88, and hence $wt(77, 88) = 0$, but $wt(77, 99) = wt(99, 88) = 3$. We can now think of the routing changes involved here as changes in link weights of the links 77-88, 77-99 and 99-88 without going into details of which exact routes changed. More specifically we can say weight change $\delta(77, 88) = 3$, since a maximum of 3 routes were lost during that event. As another example, if both P1 and P2 were withdrawn from AS 77's table from the state shown in Figure reffig:mpsetupa, then the changes would be $\delta(88, 11) = 1$, $\delta(88, 22) = 1$ and $\delta(77, 88) = 2$. Thus looking at weight changes gives us a way to capture aggregate routing changes that affect multiple prefixes.

Though fundamentally, we also weigh links by the number of routes carried, how we measure link weight change is different from [9, 10], as outlined below. In particular, we first group BGP updates into time bins of T minutes each. For each bin, we compute the maximum change in the link weight for each AS-AS link observed by each monitor.

Algorithm 1: ComputeLinkWeightChange

```
Data: BGP RIB and updates from a monitor
Result: Link weight changes seen by the monitor
begin
    foreach prefix p in RIB do
        record current path to p
    pset ← ∅
    foreach prefix p in UPDATES do
        foreach added or lost link l do
            pset[l] ← pset[l] ∪ p
            update path to p
    lwc ← ∅
    foreach link l in pset do
        lwc[l] ← member-count(pset[l])
    return lwc
end
```

For example, if $wt(a, b) = 1000$ at the start of the bin, and within the time interval has a lowest and highest value of $wt(a, b) = 900$, and $wt(a, b) = 1100$ respectively, then weight change $\delta(a, b) = 200$, thus capturing the range of link weight change during that interval. We use this measure of link weight change to capture the *effect* in terms of which links are observing route changes to how many prefixes. Note, that link weight changes are relative to the observation point or monitor, and we capture the link weight changes for multiple monitors. Next, we show how we use PCA to combine the information from multiple monitors to identify the outlier links.

3. APPLYING PRINCIPAL COMPONENT ANALYSIS

From previous section, we saw that our metric of weight change gives us an aggregate measure of the route changes on each link seen by each monitor. We argue that examining these weight changes along the original dimension of individual monitors not only presents too many dimensions to examine, but also cannot capture changes that are common across different monitors. We use Principal Component Analysis to transform the original dimensions into the new dimensional space, and show how analyzing the route changes along this space can be much more meaningful.

3.1 PCA Introduction

Principal Components Analysis (*PCA*) is a well-known statistical technique used for understanding the variance of a multi-dimensional data set. PCA maps a set of points from a n-dimensional space into a new orthogonal n-dimensional space, where the variance of the original data along each axis is maximized. The axes of the new space are called *principal components* abbreviated as PCs. The coefficients of the new reference axes are called *loadings*, and the projections of the original data onto these axes are called *scores*.

Given a $m \times n$ matrix $\mathbf{X}$, where each row represents a point, and each column represents the original dimension, PCA computes n principal components $\mathbf{v}_{i}{}_{i=1}^{n}$ defined as follows: $\mathbf{v}_k = argmax_{||\mathbf{v}||=1}||(\mathbf{X}^T - \sum_{i=1}^{k-1}\mathbf{X}^T\mathbf{v}_i\mathbf{v}_j^T)\mathbf{v}||$. The principal components are the n eigenvectors of the estimated covariance matrix and are ranked according to the amount of variance they capture in the original data. The variance of each component is described by the corresponding eigenvalue. When the input matrix is zero-mean, the first principal component contains the most variance in the original data, and any other k^{th} principal component - with $k = 2, ...n$ - identifies the maximum variance in the remaining data, i.e. the original data after removing the contributions of the previous $k-1$ components.

The original purpose of PCA was to reduce a large number of variables to a smaller number of principal components without losing much of the variance in the data. In addition to dimensionality reduction, if the resulting principal components can be meaningfully interpreted, then PCA can also provide useful insight into how original variables relate to each other.

3.2 Applying PCA on link weight changes

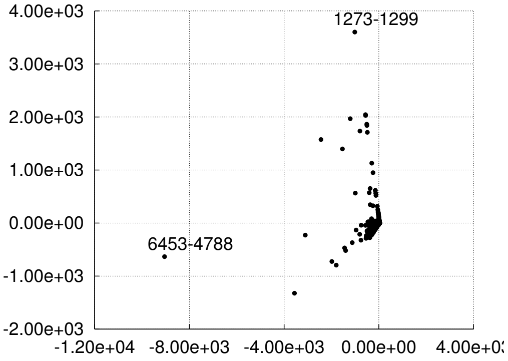
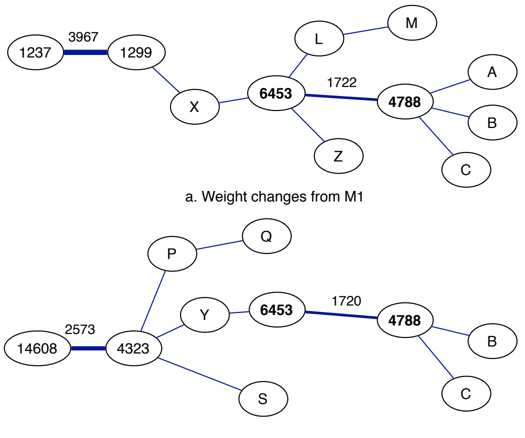


Figure 2: Outlier links from PCA output

Before delving into details, we use a simple example highlighting the need to combine information from multiple monitors. We present an example from real BGP data collected from a set of multiple monitors. If we examine the weight changes along individual monitors, we find different links having the highest weight changes. For example, the links 1237-1299 and 14608-4323 have the highest changes respectively from two sampled monitors. What we want to find out is which link stands out when information from monitors is combined together.

Without going into details, lets see how applying PCA on this data set of weight changes can help. With PCA we get a new set of components, and we project the links onto these top two components as shown in Figure 2. We can see here that the links 6453-4788 and 1273-1299 clearly stand out from the rest and can be easily identified as an outliers. While many monitors saw these links change routes, they did not stand out as the highest routing changes from that monitor. Figure 3 shows links involved in weight changes seen by two monitors M1 and M2. We can see that link 6453-4788 has a weight change of 1722 seen by M1, but the highest weight change is 3967 routes on the link 1237-1299. This is because, the weight change in 1237-1299 is also affected by changes on other links. Similar situation can be observed for monitor M2, where the link 6453-4788 is not the highest in terms of routing changes. When PCA takes the data from all monitors into account and aligns the principal components to account for the maximum variance in the data, the link 6453-4788 stands out, thus making outlier detection more meaningful. However, in general the inference of the nature of the outlier depends on the interpretation of the PC, and we explain that towards the end of this section.

More formally, to apply PCA on link weight changes, we construct a data matrix X of link weight changes seen by different monitors, where columns represent the monitors, rows represent the links, and an entry (i, j) represents the weight change on link i seen by monitor



a. Weight changes from M1

b. Weight changes from M2

Figure 3: Weight changes seen by individual monitors for case in Figure 2

j. Before applying PCA, we adjust the data so that it has zero mean. This ensures that the first principal component captures the true variance in the data.

Prior work like [25] often incorporated time as an inherent dimension for the PCA input matrix. We are primarily interested in understanding which links stand out from the rest in terms of routing changes as viewed from different monitors at a given point of time. Over time, monitors will be affected differently for different events and incorporating time in the input would make the interpretation of results more complex. Our approach is different in that we slice time into fixed size bins and analyze the bins independently of each other. By treating each time bin independently, we are able to gauge the impact of a routing change, by understanding how monitors are affected.

Next we look at some characteristics of our data set that influence what we can get out of PCA.

3.2.1 Multivariate Normalcy

Some proponents of PCA argue that the input data should follow a multivariate normal distribution. Others claim that the condition of multivariate normalcy makes the application of PCA too narrow, and that PCA can be more generally applied than that. [7] explains that if PCA is applied for descriptive purposes then it can be applied even if the data set does not follow a multivariate normal distribution. We performed tests on our data set for about 200 different randomly selected time intervals, and our results show that while some time bins show multivariate normalcy, others do not and we cannot conclusively say anything about whether multivariate normalcy exists in general. To be on the conservative side, we assume that multivariate normalcy does not exist in our data set, and we apply PCA to reduce the data set dimensionality and describe the existing variance rather than for inferences.

3.2.2 Covariance versus Correlation Matrix

In general, PCA uses two kinds of matrices as input, i.e. the covariance matrix and the correlation matrix. Correlation matrix is especially recommended when the dimensions (columns) represent measures in different units (e.g. centimeters versus millimeters) or the scales of the dimensions are very different. Using a covariance matrix in such cases will cause higher absolute values to dominate. However, since our columns are monitors with each containing a full routing table (same scale of 250k prefixes), higher absolute values (link weight changes) are important in our analysis and hence, we use a covariance matrix for PCA.

3.3 Analyzing data in New Dimensional Space

After applying PCA on the matrix described above, we get a set of principal components (new dimension) with each principal component as a linear combination of the monitors (old dimension). Our claim is that analyzing the data along the new dimension is much more beneficial than the older dimension of individual monitors. Our first task to evaluate how many PCs we need to retain. To achieve this, we use the parallel analysis technique to select the principal components that account for most of the variance.

3.3.1 Component Selecting using Parallel Analysis

There are many different techniques to decide how many components to retain. We use a technique called *parallel analysis* [15] where the obtained eigenvalues are compared to those one would expect to obtain from random data. If the first m eigenvalues are those which have values greater than what would be expected from random data, then one adopts a solution with m factors.

3.3.2 Outlier Link and Change Magnitude

Once, we decide to retain a set of principal components, the resulting data set consists of each point represented by k co-ordinates, where k is the number of components retained. Our goal is to identify outliers in this representation. In general for $k-variate$ data, the definition of *outlier* implies that the points are a long way from the rest of the observations in the k-dimensional space. Since each principal component represents a linear combination of the original monitors, we examine outliers by looking in the directions of the selected principal components.

We define the *outlier link* as the link that is farthest away from the origin when projected on a principal component. We find outlier links for each of the selected principal components, since different principal components may capture potentially different routing effects. Note, we use a very simple outlier link detection technique, and we realize that we may capture more information by capturing more than one outlier link based on distances from the rest of the links. However, our primary aim here was to explore this direction of analysis to see if its beneficial and then use more sophisticated outlier detection techniques during the next stage.

Note, also that each link has a score assigned to it along each principal component, representing the point's coordinates in the new dimensional space. For example, in Figure 2, the link 6453-4788 is an outlier on PC1 and the link 1273-1299 is an outlier on PC2. The coordinates of the links on the PCs are called scores. We define *change magnitude* of a principal component as the score of the outlier link on that component. Note that score values can be negative or positive, and as such the signs are completely arbitrary based on which direction the component goes. We consider absolute value of the score as the change magnitude.

3.3.3 Interpreting the Principal Components

In addition to observing outlier links on a principal component, we also need to interpret the principal components. Each principal component captures some variability in the data, and hence its important to understand the interpretation of each PC in order to put the observations on the principal component in context of the original monitors. Note, that the sign of any PC is completely arbitrary, and what matters in the interpretation is the relative signs across different coefficients. Also, if one considers all coefficients including the ones with really low value, then the interpretation of the PC can be difficult. One way to simplify as suggested in [7] is to consider values above a quarter of the highest absolute value. We call the coefficients thus selected as *chosen coefficients*. Broadly, there are two classes of principal components based on the relative signs of the monitors coefficients.

1. Uniform Effect: Here we observe the chosen coefficients of the principal components to have the same sign, either positive or negative. The coefficient value of each monitor indicates the amount of influence that particular monitor has on the component and with multiple monitors influencing the PC, there must be link changes that are common to these monitors.

2. Contrasting Effect: Here we observe the influential coefficients of the principal components to have opposite signs. This indicates a contrast in the observations of sets of monitors, with the PC aligning somewhere between the directions of the two sets of monitors, ones with positive coefficients, and ones with negative coefficients.

Clearly, interpreting the components is much easier when we observe a uniform effect. In these cases, by observing the monitors corresponding to the chosen coefficients, one can estimate how many monitors influence the PC, and hence establish how global the effect of the changes on the selected outlier link. The interpretation is trickier with contrasting effect, and this usually occurs in the presence of multiple independent events of similar magnitude in the same time interval, and thus the contrasting effect on the principal component. In the vast majority of our cases, the principal components fall into the uniform effect category.

4. VALIDATION THROUGH SIMULATIONS

In this section, we use Internet scale simulations to provide a basic validation of our scheme. In particular, we are interested in knowing if the outlier links as captured by our scheme involve heavy routing changes.

4.1 Setup and Route Computation

For our simulations, we use an AS topology inferred from BGP routing tables and updates, representing a snapshot of the Internet as of Feb 2006 (available from [26]). The details of how this topology was constructed are described in [27]. The topology consists of 22,467 AS nodes and 63,883 links. We classified each link as either customer-provider or peer-peer using the PTE algorithm[5] and used the *no valley prefer customer* routing policy to infer routing paths. We randomly picked 30 nodes and designated them as monitors. We can observe the routes from these monitors to all destinations. The resulting monitor set represents a set of monitors mostly from the edge of the network, and does not share high path overlap among themselves.

We modeled each AS as a single node and used the routing tables collected from RV [18] from the time of topology snapshot to get a mapping of how many prefixes were announced by each origin AS at that time. After this step we had a total of about 180,000 prefixes announced. We abstracted the router decision process into the following priorities (1) local policy based on relationship, (2) AS path length, and (3) lowest ID tiebreaker. We applied our decision process to compute the routes from each monitor to all prefixes in the topology and record these routes as the initial set of routes. A similar setup and decision process was used in previous works such as [4]. Later work [16] reported the inaccuracies in path predictions resulting from abstracting AS as a single node. However, such a setup is still useful for basic validation and can provide an important sanity check.

4.2 Simulating Routing Events

Once the initial set of routes was computed, we simulated various problem scenarios, recomputed the new set of routes, computed link weight changes and applied PCA on this data set. Note that we do not simulate the propagation of BGP updates in the network, but rather remove links from the topology and recompute the routes. As a result, in this setup, we do not have to worry about timing issues. Instead we have the set of initial routes, a set of final routes, and can compute the resulting link weight changes. We understand that the possibility of multiple peerings between ASes means that our setup of link failure where the link vanishes completely is a simplification. In order to pick links to fail, we computed the weight of links by counting the number of routes that every monitor routes through it.

We simulated two kinds of events described below.

Local link event In this case, we failed heavy weight links mainly seen by one monitor. We failed a total of 80 heavy weight links.

Non-local link event We failed links with heavy weight that are seen by multiple monitors a single link between two tier-1 ASes. We picked the top 20 most used links in addition to the top 10 tier-1 links, and failed a total of 30 links.

Note, one may argue that detecting local events on a single monitor does not require PCA. However, the fact is that in BGP data, local events do occur, and its important that we know whether our technique is able to prove that they are indeed local. Also, our goal is to not detect link failures, rather to detect outliers in terms of most routing changes from multiple monitors. Since we fail a link, that link will observe lots of route losses, and this is useful for our validation.

4.3 Simulation Results

We now present the result of our PCA analysis on the link weight changes seen by our set of monitors.

4.3.1 Local link events

The local link events are more straightforward than the global link events. In all the cases, the failed link appeared as the outlier link. Importantly, we found that the first principal component was strongly influenced only by the monitor adjacent to the failed link.

4.3.2 Global link events

We found that in all the cases, the failed link appears as the most variant link on the first principal component. We also found that the first principal component is influenced by more than one monitor indicating as discussed in Section 3.3.3 a uniform effect seen by multiple monitors.

As a representative we discuss the failure of the link between 1273 and 6663. Figure 4 shows the score plot with the first 2 principal components with the X axis being the first principal component and hence where maximum variance is captured. We can see from this plot that the link 1273-6663 is the outlier link on the first principal component, as it is farthest away from the origin. The PC1 also shows a uniform effect with all coefficients as negative and the top-5 coefficients being $\{-0.28, -0.27, -0.25, -0.23, -0.23\}$ suggesting that multiple monitors influence this component. Notice, the second component has 702-12301 as an outlier, but unlike PC1, on analyzing the PC2, it shows a heavy influence of one monitor on the PC.

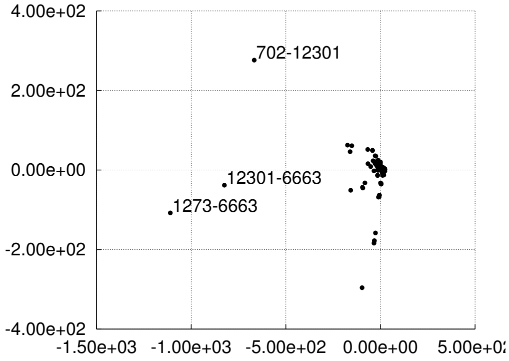


Figure 4: Outlier link seen via PCA

Summary

The simulation results show that the failed links show up as outliers in our analysis after applying PCA. Also, the scope of the failure is reflected in the interpretation of the PCA based on the coefficients of the monitors. We would like to stress that the aim of the simulation setup is not to see whether we can capture failed links or not, but rather to provide a basic validation of our scheme to check whether the links that appear as outliers indeed have lots of route changes. As it happens, the failed links lose most routes in our setup and hence appear as outliers.

5. DATA SET PREPARATION

We prepare PCA input data by computing link weight changes from BGP updates collected from multiple monitors across the Internet. We now go into details of how we collect and process the BGP updates in our approach.

5.1 BGP Monitor Selection

When applying PCA on link weight changes, its important to avoid two or more monitors seeing very similar changes because of their connectivity. For example, if two monitors have the same providers, then its very likely that they will observe common dynamics. Thus, before we apply PCA, we carefully select a set of monitors that represent as diverse views as possible.

As of October 2007, there are 557 monitors available from RouteViews and RIPE. We start by looking at routing tables from 3 routing tables spanning different dates in the month of May 2007. Before we apply our algorithm, we first prune out monitors that belong to the same AS or that do not export full routing tables [1], leaving us with a set of 80 monitors. Specifically we use two metrics to evaluate the monitor set.

- Neighbor ratio of M_1 with M_2: Indicates what fraction of M_2's routing table uses M_1 as the immediate next hop.
- Sibling ratio of M_1 with M_2: Indicates what fraction of M_1's routing table shares the same next hop with other monitors.

Thus, imagine a case where monitor M_1 is a provider of another monitor M_2. The neighbor ratio will be high for M_1 if M_2 uses M_1 as a next hop to reach a high percentage of the prefixes. The neighbor ratio is intentionally defined this way to prefer monitors towards the edge instead of the core.

We construct a neighbor ratio matrix, $\mathbf{R}_n$, where the value in the matrix, $\mathbf{R}_n^{i,j}$ at row i and column j, is the direct ratio of mon_i with mon_j. Each value in the matrix has the maximum value of 1 in the case where all prefix entries of mon_i has mon_j as the next hop and the minimum value of 0 when none of the prefix entries of mon_i has mon_j as the next hop. We sum up all $\mathbf{R}_n^{i,j}$ values across the rows to obtain view-overlap, $vo_j = \sum_{k=1}^{n} \mathbf{R}_n^{k,j}$, values which represent the overall view overlap for mon_j in terms of every other monitors in the set.

From the matrix $\mathbf{R}_n$, we iteratively eliminate mon_j with highest vo_j until the highest value in $\max(vo_{1..n})$ is less than a threshold. Note, after each removal, we have to recompute the matrix, since ratios change. We use a threshold of 0.1, indicating that in our residual set no two monitors have a neighbor ratio of more than 0.1. After this step, we are left with a set of 62 monitors.

Computing the sibling ratio is done in the similar fashion by constructing a matrix $\mathbf{R}_s$, and the summation of the rows indicates the sibling overlap for each monitor. We then iteratively remove monitors with high sibling overlap. Figure 5 shows the residual max sibling ratio after each removal. We carry out the removal until the threshold of 0.3, is reached indicating that no two monitors have a sibling ratio higher than 0.3. After the sibling removal we are left with a set of 30 monitors that we use for results. We verified that this selected set of monitors cover a large enough portion of the Internet by checking for the presence of tier1-tier1 links as well as other well known links between large ISPs.

5.2 BGP Data

We analyzed the BGP routing updates from the set of 30 monitors for a 12 month period from Jan 1, 2007 to Dec 30, 2007. Over this period, there were some known outages with RouteViews collection boxes and we believe these outages only result in data missing for short periods and do not affect our results in a major way.

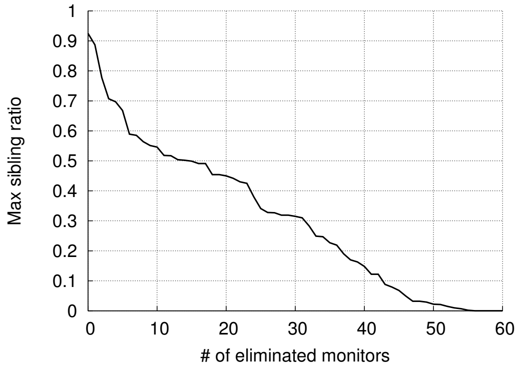


Figure 5: Max sibling ratio

We use a time interval of 10 minutes to group updates together. This time interval is small enough to reduce the probability of multiple events occurring in the same time interval, and at the same time long enough to allow BGP routes to converge. Prior work has suggested that most cases of BGP routing convergence due to the same do not last more than 5 minutes [8]. Other related work using BGP updates to identify network wide disruptions [6] has also used a similar value of 10 minutes to group updates into bins. For each 10 minute period, we applied Algorithm 1 to compute the link weight changes for each of the 30 monitors on each time bin.

6. RESULTS

We now present results from the application of our scheme on BGP data over a period of one year. We first identify large scale routing instabilities and investigate what kinds of events can cause such large scale routing instabilities. In the later part, we identify routing instabilities that repeat over long periods of time.

6.1 Large Scale Events

Recall from Section 3 that we define *change magnitude* on a principal component as the variance of the outlier link on that principal component. In principal, the higher the magnitude the bigger the event as seen in the new sub-space. So our first task is to identify a set of high magnitude events from all the events. Ideally we would like to compare the change magnitudes across different time intervals to separate out small events from big events. However, since we apply PCA independently on each time bin, one has to be careful while comparing change magnitudes across different time slots. We discuss this next.

[1] Some BGP routers export only partial routing tables to the data collectors

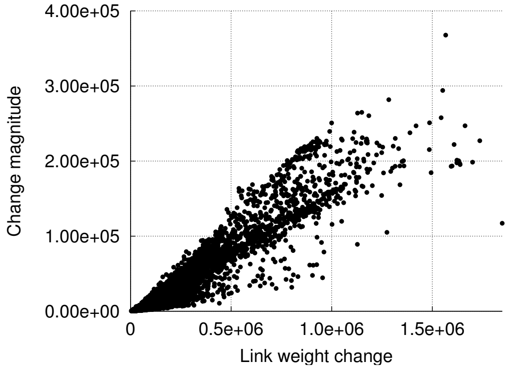


Figure 6: Link weight changes versus Change Magnitude

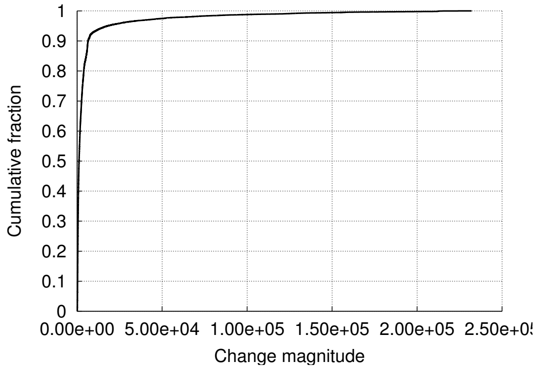


Figure 7: CDF of change magnitude

6.1.1 *Separating High Magnitude Events*

In order to better understand how comparable the change magnitudes are, we first study the correlation between the change magnitude and the link weight changes that serve as input to our scheme. Figure 6 shows the correlation plot with X axis as sum of link weight changes for all monitors in a time slot, and the Y-axis as the sum of the change magnitudes on the top 10 components. Note, typically about 1-3 components capture most of the variance in the data, but we use 10 components to be on the conservative side, even though from our observation, the last few components only marginally add to the variance.

We found that the correlation coefficient for Figure 6 to be 0.93. Note, that the sum of weight changes is a collective effect of lots of routes changing, and a certain number of monitors observing the change. Our objective in presenting Figure 6 is to establish a trend that high weight changes generally result in high magnitudes, and this is used just to help us draw a line to separate large events from smaller events.

For each time bin over the one year period, we pick the change magnitude of outlier link on the top component to represent the change magnitude of that bin. Figure 7 shows the cumulative distribution of the change magnitudes for all the bins. The increase in magnitude is fairly linear until about $y = 0.9$ where curve changes shape. We use this point as the cutoff and treat all the change magnitudes above this point as high magnitude events. This gives us a total of 5310 bins to investigate for high magnitude events. We now study these 5310 bins in more detail to understand the scope of the high magnitude events, namely what percentage of the Internet is impacted by these events.

6.1.2 *Identifying Events with Local scope*

Defining scope of a large scale event is not trivial and there are two main issues. First, even for a single prefix, we do not have access to BGP routers in every AS, and can at best estimate what percentage of the Internet might have been affected by looking at the sample of monitors involved. Second, during a routing event involving multiple prefixes, each monitor may see changes to a different number of prefixes, thus making simple counts difficult.

We follow a two level procedure to scope. First we classify events into purely local and non-local. Then, we concentrate on understanding the distribution of events in the non-local category. Recall that each PC is a linear combination of original monitors, and by looking at the load values of original monitors on a component, we can see which monitors are influencing the component and by how much. For the first level of classification, we start by looking at the highest load value on the first component, representing how influential is the most influential monitor on that component. Figure 8 shows the distribution of the top 2 monitors influencing the first principal component. We see that for over 95% of the events, the highest load value is very close to 1, and the 2nd highest load value on the first principal component is less than 0.1, indicating that the component is influenced very strongly by a single monitor. We treat all events above $x = 0.9$ in Figure 8 as local events. Thus, we can say that most high magnitude events are local in scope, i.e. occurring close to one monitor but not affecting other monitors. We investigated some of these local events by looking at routing updates and found that it typically involves an AS switching lots of routes from one provider to another provider.

By looking at events below $y = 0.9$, we are thus left with a total of 126 events that are non-local in scope and we examine these cases in detail next.

6.1.3 *Non-local High Magnitude Events*

For events classified as non-local, we first establish a way to understand the scope of the event. We are primarily interested in trying to establish how many monitors from our set of monitors are impacted by the route change seen on the outlier link. Recall that each princi-

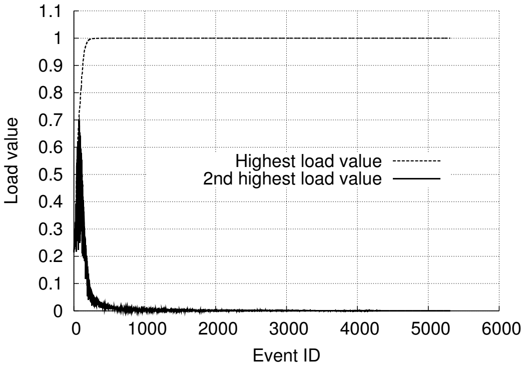


Figure 8: Load values of top 2 monitors on first component

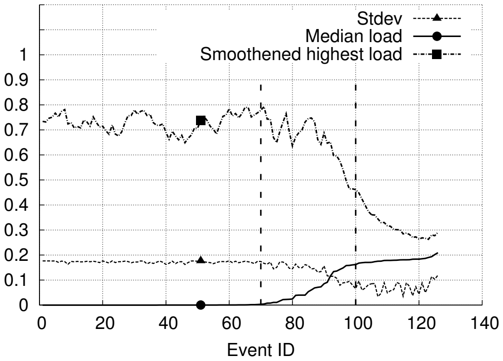


Figure 9: Non-local events

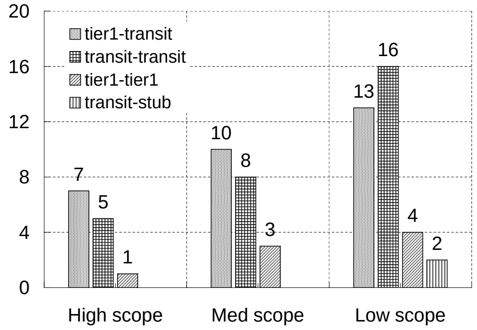


Figure 10: Link types in non-local events

pal component is a linear combination of monitors. For each principal component, we identify the median load value of the monitors and the standard deviation. The median load value indicates the typical influence of a monitor on that component. By examining the standard deviation, we can gauge how much the influence differs among all the monitors. Figure 9 shows the median load values of the monitors, along with standard deviation. To show the trend of changes in the highest load value for these events, we also include the exponentially smooth highest load value in Figure 9.

Based on Figure 9, we break down the non-local events into three categories. From $0 < x < 70$, the median is very close to 0 and standard deviation is close to 0.2. We call these class of events *low scope* non-local events. From $100 \leq x \leq 126$, we can see the median values close to about 0.2, and the standard deviation around 0.1. We call this region as the *high-scope* non-local events. Finally, we call the events in the area in between from $70 \leq x < 100$ as *medium scope* non-local events. When we look at the unique links appearing in this set of 126 events, we observe that a lot of links appear twice and this is because the first appearance is when there is a problem, while the second appearance is when things return to normal. Next we see where in the topology these links appear.

To categorize the outliers, we use a simple classification of AS nodes based on connectivity. We use the AS link information collected from BGP routing tables and updates, available from [26]. We only consider ASes seen between Jan 1, 2007 and Dec 31, 2007. We classify AS nodes into three tiers: Tier-1 nodes, transit nodes, and stub nodes. To choose the set of Tier-1 nodes, we started with a well known list, and added a few high degree nodes that form a clique with the existing set. Nodes other than Tier-1s but provide transit service to other AS nodes, are classified as *transit* nodes, and the remainder of nodes are classified as *stub* nodes. Figure 10 shows the types of links involved in each of the three bins. For the high scope events, most of them involve new prefixes, so in some sense we cannot read much into which links are involved. For the medium and lower scope events, we observe that tier-1 links (between themselves or to transits) are involved in many of the cases.

6.2 Case Studies of Non Local Events

We now investigate the events that caused global disturbance by looking at BGP updates. We find that except for one, all the other events in the high scope category of non-local events involve new prefixes being announced by one end of the outlier link.

6.2.1 Involving New Prefixes

First, we analyzed the set of prefixes whose route changed during each event. Then, we classified the events into the following categories. Given a set of prefixes P_1 usually announced by AS_x, when AS_x announces a set of prefixes P_2, s.t. $|P_2| >> |P_1|$ for a limited time interval, we identify

Announcement of deaggregated prefixes, if P_2 covers (almost) the same prefix space as in P_1.

Announcement of new uncovered prefixes, if there is (almost) no overlap in the address space between prefixes in P_1 and prefixes in P_2. This category includes events occurring when ASes announce their "private" address space, likely because of some misconfigurations.

Table 1 describes the events analyzed, along with their categories.

AS-link	Count	Origin AS	category
7018-7015	4	7015	new uncovered prefixes
2200-3356	3	3356	
3549-11456	2	11456	
1237-2200	1	2200	
28513-8151	1	8151	
6453-4788	2	4788	de-aggregation
7018-4788	1	4788	
3257-5486	2	5486	
1239-209	2	209	
17622-9394	1	9394	
7018-33650	1	33650	

Table 1: Summary of cases involving new prefixes

Note that all links that appeared only once were cases where the original state was restored in the same time bin, i.e. lasting less than 10 minutes.

6.2.2 Link Problems causing Global Disturbance

Besides the cases involving new prefixes, we observed one case where a link problem caused global disturbace. On September 19, 2007, around 17:00 GMT, all monitors observed over a thousand routes switching away from the link 3356-6395 to different alternate links. Again, we see that most monitors see this change because they use this link to reach the prefixes announced by AS 6395, and hence are affected.

For other links in the medium and low scope category, where a portion of the monitors are affected, in most cases, the affected set includes the monitors somewhere in the customer tree of the two transit ASes. Depending on the connectivity and type of relationship (i.e. customer or peer), the scope varies. We analyzed AS paths from routing tables and found that very few links carry a heavy weight as seen by a majority of monitors, and such links are usually ones that are lower in the topology tier hierarchy (e.g. a large regional ISP connecting to global ISPs). This explains why we do not see a truly global disturbance due to link instabilities.

Summary

Our results here show that events that have truly global impact usually involve announcements and withdrawals of prefixes carried out by the origin AS or its provider. On the other hand, events where link instabilities affects transit routes are usually constrained to a smaller number of monitors, somewhere along the customer base of the end points of the link.

6.3 Repeated Routing Instabilities

Having analyzed the events along the magnitude dimension and examining the highest magnitude events, we now look at the events along the frequency dimension. In particular, we are interested in identifying links that appear repeatedly as outliers. Instead of looking at all the link events, we consider events above $y = 0.5$ in the distribution of change magnitude shown in Figure 7. Since we run PCA on each time bin, we want to avoid bins where very few aggregate routing changes were observed. For the selected events, we find out how many times a link appeared as an outlier. Figure 11 shows the distribution of the number of appearances of a link as an outlier. Clearly there are a few links that appear quite often as outliers. We use a threshold of $y = 0.8$ corresponding to $x = 15$ to separate the more frequently appearing from the lesser one. We have a total of 212 links with $x > 15$ appearances.

We now investigate the scope of the appearance of these outlier links, i.e. whether they are seen by small set of monitors or globally observed. Recall from our analysis of high magnitude events, that for each appearance, we use median load value and standard deviation. Each appearance of a link could potentially have different scope. To understand the scope over multiple appearances for a link, we assign the link scope as the median of the median load values over all the appearances for that link, and the standard deviation of all the median load values. Thus, a low standard deviation indicates that the median value is more or less consistent over all the appearances. Figure 12 shows the distribution of the computed link scope with smoothened standard deviation to show the trend. We can see that almost all the repeated outliers have very low scope. We are interested in the small few that cause a more global disturbance. Based on the link scope, we study events above $x = 190$. Figure 13 shows the types of links involved in the low and high scope categories. We can clearly see here that links between tier-1s are quite stable in general and observe only 5 instances where they appear more than 15 times, and they show low scope. We now examine the most global repeated appearances to understand where they occur.

6.3.1 AS 4637-AS 4761

The link between AS 4637 (Reach) and AS 4761 (Indosat) appears as an outlier 31 times from January 2007 to Febrary 2007 with the highest scope among repeatedly appearing outlier links. By examining BGP data, we found that the routes using link 4637-4761 switched to using either to 1239-4637, 7473-4637, or 3491-4637 for about 10 to 30 minutes. This phenomenon was observed from January 2007 to Febrary 2007 only, and from March 2007 to December 2007, the link never shows up as an outlier again.

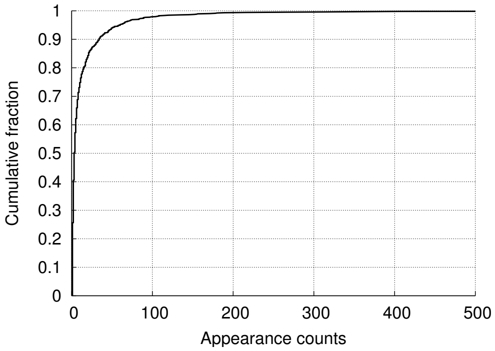


Figure 11: Repeated appearances as an outlier link

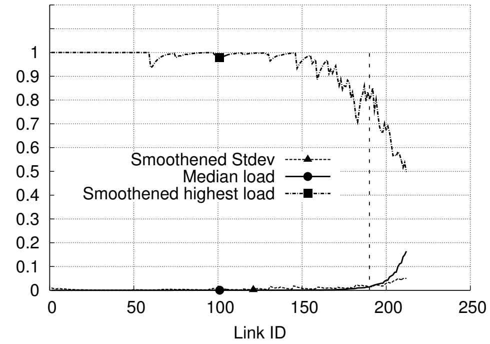


Figure 12: Understanding scope of repeated events

6.3.2 AS 6453-AS 30890

The link between AS 6453 (Teleglobe Inc) and AS 30890 (Evolva Telecom) appears 83 times throughout the one year period with the second highest scope in our repeatedly appearing outlier link set. We found that about 500 routes to AS 30890 or using AS 30890 as an intermediate node in AS-PATH switched to the alternate longer route 6453-5588-5606-30890 for the instability duration of about 10 to 30 minutes.

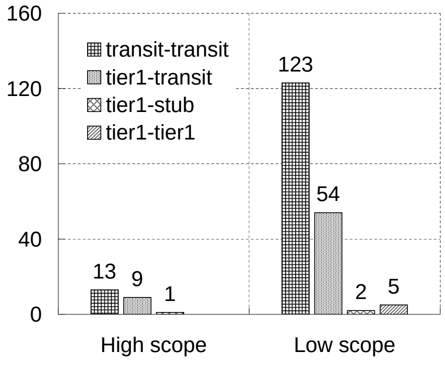


Figure 13: Link types in repeated events

Summary

We find many cases of links appearing repeatedly as outliers. However, most of the repeatedly appearing links have a low scope. In the few cases where the effect is more widely observed, some have a short bursty period of instability, while others show up more routinely as outliers throughout our study period. Further, the duration of the change usually lasted about 30 minutes or less, indicating that this was less likely to be intentional changes.

7. DISCUSSION

The Internet routing infrastructure is a large scale distributed system, not only it maintains the routes to hundreds of thousand destination prefixes, but also different monitors have different views on the routing state. The sheer scale of the system makes assessing the overall Internet routing stability a difficult problem. In this paper, we tackle this problem with two new approaches, the use of link weight to measure the aggregate routing changes over each AS link, and the use of PCA to process link weight changes as seen from multiple monitors.

The outcome of PCA enables us to gauge each routing event's effects on all the monitors, so that we can identify the *scope of impact* of each event. This result sets our work apart from previous efforts on routing dynamics measurements like [2] , where one is interested in the *cause* or *location* [4] of observed routing stabilities. Our effort also differs from previous work in applying PCA to BGP data analysis [25], where only data collected from a *single* monitor is used.

We would also be the first to admit that, in taking the first step towards gauging the global routing stability, our results are preliminary and a number of issues need to be further explored. First, we would like to refine our use of link weight changes in the analysis. Recall that we find the absolute value of the maximum weight change seen by a link in each time bin, without knowing whether routes were lost or gained. We plan to investigate the use of signed weight changes, which can show whether a link appears as an outlier with routes gained or lost. This information may help in the analysis of events, especially when doing temporal correlation on the same link: Observing an outlier link with a negative change followed shortly by the same link with a positive change can be looked at as a failure followed by recovery or vice versa. Of course we must be aware of the possibility that such failure and recovery cycle may occur within the same time bin, thus the signed weight changes could cancel each other, leading to a missed event.

We would also like to acknowledge the limitations of our basic outlier detection scheme. Note, that we only choose one link per component, and there could be links which are very close to the outlier link. This can happen especially when the links involved in routing changes have a chain like topology with similar routing changes. This can also occur when the event is purely local and close to an observation point. We are currently investigating alternative methods for picking outlier links based on distance estimation. Nevertheless, even looking at a single outlier link for each principal component gives us a good start to understand Internet routing stability.

Finally, we believe it is important to be able to reproduce research results and enable others to be able to compare our approach with others. To this end, we have made our scripts and a small part of our data set publicly available [1] in order to make our results completely reproducible.

8. RELATED WORKS

8.1 Identifying origins of routing changes

In a seminal works regarding network instability, Labovitz *et. al.*[8] identifies several causes of routing instabilities in the Internet, without however diagnosing their topological origin. Later efforts [3, 2, 4, 24] analyze BGP updates by aggregating data along one of the three dimensions: time, monitors and prefixes, to obtain the candidate sets of routing instability origins. Both [4] and [10] identify the origin of the routing instabilities, the former adopts a greedy approach based on removing links, while the latter uses a min-cut on a flow graph connecting links involved in the changes. Teixeira *et. al.*[22] describe a framework to detect the cause of a routing change using a coordinated diagnostic mechanism among several ISPs, requiring a special server in each ISP that replies to diagnose queries from other domains.

8.2 Applying PCA to Internet data analysis

Principal Component Analysis has been applied to both traffic and routing information to help understand network dynamics. [12, 11, 14, 13] first proposed an approach based on PCA for detecting volume anomalies in traffic data collected by several monitors within a network. A volume anomaly denotes unusual traffic load levels. They observed that, although traffic data is high-dimensional (in terms of number of links), normal traffic patterns are intrinsically low-dimensional. Thus, they separated network traffic into a normal subspace, and an anomalous subspace, and they used the minor components of PCA to identify volume anomalies. [21] showed that tuning PCA to operate effectively is nontrivial. [28] introduced the *temporal PCA*, which exploits temporal correlation to identify dominant pattern across time. In contrast, [12, 11, 14, 13] analyzed the correlation between traffic on different links (*spacial PCA*).

The work that is most relevant to ours is by Xu *et al.* [25] who applied PCA to routing data to analyze Internet-wide events. Given a stream of BGP updates collected by a *single* monitor over time, they group prefixes that are likely affected by the same network event. [6] focused on diagnosis of network disruptions within a single network, and used PCA to combine multiple BGP updates streams coming from distinct observation points. Both [25, 6] used time-series matrices, which count the number of BGP updates received by a single router in each time slot. In contrast, our work exploits the topology dimension. For each given time slot, we measure the number of routing changes over each link or AS as observed by multiple monitors.

Overall, applying techniques like PCA on routing data is a relatively unexplored direction, and ours is the first that applies it on an aggregate metric using multiple monitors.

9. CONCLUSIONS AND FUTURE WORK

Due to complex interconnects among ASes and private routing policies, different vantage points in the Internet routing infrastructure observe different routings changes, creating a challenge in utilizing data from multiple vantage points to measure the scope of routing changes. In this paper we first capture aggregate routing changes using link weight changes, and then use Principal Component Analysis to combine information from multiple vantage points. Our work is the first of its kind that analyzes routing stability of the Internet as a whole, rather than the stability of individual routes, using information from multiple monitors and over a long period of time.

Using our scheme we found a number of instances of routing changes that were observed within a large scope. We further discovered that these routing changes with global effect were usually caused by the appearance of new prefixes, which were caused by either route leakages or de-aggregation. We also found that link instabilities do not cause a disturbance throughout the Internet in general. Our analysis of the AS paths shows that very few links carry a heavy weight as seen by a majority of monitors, and such links are usually ones that are lower in the topology tier hierarchy (e.g. a large regional ISP connecting to global ISPs). Overall, our measurement results over a one year period show that while routing changes do occur frequently, their scope is usually limited to a small portion of the global Internet. This may be attributed to the increasingly meshy nature of the AS topology. At the same time, this observation raises a new question: if most routing changes are confined to local scope, and routing events of global scope occur infrequently, then what may be the causes of the high volume routing updates that one frequently observes?

We plan to look into this problem next.

10. REFERENCES

[1] BGP. Internet Routing Stability Project. http://sourceforge.net/projects/bgpstat/.

[2] M. Caesar, L. Subramanian, and R. Katz. Root cause analysis of Internet routing dynamics. Technical Report UCB/CSD-04-1302, U.C. Berkeley, november 2003.

[3] D. Chang, R. Govindan, and J. Hiedemann. The temporal and topological characterestics of BGP path changes. In *ICNP*, november 2003.

[4] A. FeldMann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs. Locating Internet routing instabilities. In *Proceedings of Sigcomm*, September 2004.

[5] L. Gao. On inferring autonomous system relationships in the Internet. *ACM/IEEE Transactions on Networking*, 9(6):733–745, 2001.

[6] Y. Huang, N. Feamster, A. Lakhina, and J. Xu. Detecting Network Disruptions with Network-Wide Analysis . In *Proc. of ACM SIGMETRICS*, 2007.

[7] I. T. Jolliffe. *Principal Component Analysis.* Springer-Verlag, 2002.

[8] C. Labovitz, G. R. Malan, and F. Jahanian. Origins of internet routing instability. In *Proceedings of the IEEE INFOCOM '99*, pages 218–26, New York, NY, 1999.

[9] M. Lad, D. Massey, and L. Zhang. Visualizing Internet routing changes. In *IEEE Transactions on visualization and Computer Graphics, special issue on visual analytics*, to appear, 2006.

[10] M. Lad, R. Oliviera, D. Massey, and L. Zhang. Inferring the Origin of Routing Changes using Link Weights. In *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2007.

[11] A. Lakhina, M. Crovella, and C. Diot. Characterization of Network-Wide Anomalies in Traffic Flows. In *Proc. of Internet Measurement Conference*, pages 201–206, New York, NY, USA, 2004. ACM Press.

[12] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-Wide Traffic Anomalies. In *Proc. of ACM SIGCOMM*, pages 219–230, New York, NY, USA, 2004. ACM Press.

[13] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *SIGCOMM Comput. Commun. Rev.*, 2005.

[14] A. Lakhina, K. Papagiannaki, M. Crovella, C. Diot, E. D. Kolaczyk, and N. Taft. Structural Analysis of Network Traffic Flows. In *Proc. of ACM SIGMETRICS*, pages 61–72, New York, NY, USA, 2004. ACM Press.

[15] R. Montanelli and L. Humphreys. Latent roots of random data correlation matrices with squared multiple correlations on the diagonal: A monte carlo study. *Psychometrika*, 41(3):341–348, September 1976. available at http://ideas.repec.org/a/spr/psycho/v41y1976i3p341-348.html.

[16] W. Mühlbauer, A. Feldmann, O. Maennel, M. Roughan, and S. Uhlig. Building an AS-topology model that captures route diversity. In *Proceedings of ACM SIGCOMM*, 2006.

[17] R. NCC. Routing Information Service. http://www.ris.ripe.net/.

[18] U. of Oregon. Route Views Project. http://www.routeviews.org.

[19] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol (BGP-4). Request for Comment (RFC): 4271, 2006.

[20] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang. BGP routing stability of popular destinations. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop 2002*, 2002.

[21] H. Ringberg, A. Soule, J. Rexford, and C. Diot. Sensitivity of PCA for traffic anomaly detection. In *Proc. of ACM SIGMETRICS*, 2007.

[22] R. Teixeira and J. Rexford. A measurement framework for pin-pointing routing changes. In *Proceedings of the ACM SIGCOMM workshop on Network troubleshooting*, 2004.

[23] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Observation and analysis of bgp behavior under stress. In *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 183–195, New York, NY, USA, 2002. ACM Press.

[24] J. Wu, Z. M. Mao, and J. Rexford. Finding a needle in a haystack: Pinpointing significant BGP routing changes in an IP network. In *Proceedings of 2nd symposium on Networked Systems Design and Implementation (NSDI)*, 2005.

[25] K. Xu, J. Chandrashekar, and Z.-L. Zhang. A First Step Towards Understanding Inter-domain Routing. In *Proc. of ACM SIGCOMM Workshop on Mining Network Data*, 2005.

[26] B. Zhang, R. Liu, D. Massey, and L. Zhang. Internet Topology Project. http://irl.cs.ucla.edu/topology/.

[27] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet's AS level topology. In *ACM Sigcomm Computer Communication Review*, 2005.

[28] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network Anomography. In *Proc of the Internet Measurement Conference*, pages 30–30, Berkeley, CA, USA, 2005. USENIX Association.