Principal Component Analysis of BGP Update Streams

Kuai Xu, Jaideep Chandrashekar, and Zhi-Li Zhang

Abstract: **In this paper, we propose a novel methodology to identify border gateway protocol (BGP) updates associated with major events—affecting network reachability to multiple ASes—and separate them (statistically) from those attributable to minor events, which individually generate few updates, but collectively form the persistent background noise observed at BGP vantage points. Our methodology is based on principal component analysis, which enables us to transform and reduce the BGP updates into different AS clusters that are likely affected by distinct major events. We demonstrate the accuracy and effectiveness of our methodology through simulations and real BGP data.**

Index Terms: **Border gateway protocol (BGP) updates, principal component analysis (PCA).**

I. INTRODUCTION

Given the critical nature of the Internet routing infrastructure, understanding border gateway protocol (BGP) routing dynamics and the underlying "root causes" is crucial, but at the same time, very challenging [1]. In the recent years, several efforts have been directed at the root cause analysis of BGP updates [2]–[11], with the goal of locating where routing instabilities occur. While these efforts have made significant advances, the take-away message that is underscored is that BGP root cause analysis is extremely challenging. Several aspects of inter-domain routing complicate this task and make it very hard: First, the autonomous system (AS) paths carried in BGP updates are highly abstracted, hiding many important connectivity details, making it hard to accurately pinpoint the exact location of an event. Also, specific routing policies may cause the routing updates to obfuscate and hide the actual events [12]. Second, different events may trigger similar types of updates, making it hard to distinguish the events based on information in the updates. Third, given the size of the Internet, events are likely to occur concurrently; thus the observed updates from these events may be interleaved at a vantage point, further complicating the identification of root causes.

In this paper, instead of tackling the problem of BGP root cause analysis directly, we attempt to answer a fundamental question to the understanding of BGP routing dynamics: Based on a stream of BGP updates observed at one or more vantage points, is it possible to identify and separate updates that are likely triggered by distinct events? While addressing this question, we are particularly interested in distinguishing between updates caused by "major" network events—those that trigger a large number of updates and affect reachability to many ASes—from those that can be attributed to "minor" events that individually contribute few updates, but collectively form the BGP "noise" observed at vantage points. We hope that by identifying and separating BGP updates caused by major events, we can reduce the "noise" in the BGP updates associated with the events, and thereby facilitate the task of root cause analysis. In the following we further motivate the question raised above and outline the methodology that we propose to address this question.

We propose a novel methodology to statistically identify and separate BGP updates triggered by major events, even when they overlap with those caused by other (minor) events. Our methodology is based on *principal component analysis* (PCA), a well-known statistical method for multivariate data analysis [11]. Using PCA, we exploit the temporal correlations in the update streams to extract clusters of origin ASes whose prefixes are likely affected by the same events. Using these (origin) AS clusters, we perform "spatial correlation" and "type-of-change" analysis to further validate and corroborate our findings. We show that in most cases, the (origin) ASes within each AS cluster exhibit strong common features (e.g., with shared providers or their associated updates having similar type of changes).

The contributions in this paper can be summarized as follows:

- We propose a novel methodology to infer distinct (major) events from BGP update streams and separate likely updates associated with these events. In addition, we verify its effectiveness and accuracy using simulations.
- The work that we present in this paper significantly advances our understanding of BGP routing dynamics. In particular, we show that correlated events occur quite frequently, which is contrary to what is assumed in most efforts in root cause analysis.
- The results presented in this paper can serve to *inform* and *guide* algorithms used to perform BGP root cause analysis and trouble-shooting.

The remainder of this paper is organized as follows. Section II briefly discusses BGP operations and gives an overview of PCA. In Section III, we describe the methodology. Section IV demonstrates the accuracy and effectiveness of our methodology through simulations. Section V presents the results of our analysis on real BGP update streams. We discuss related work in Section VI. Finally, Section VII concludes the paper.

II. A QUICK PRIMER ON BGP AND PCA

A. Border Gateway Protocol

BGP is an *incremental, path-vector* protocol. In other words, once a session is established between neighboring routers, route updates are exchanged only in response to routing events. Suppose a session between a pair of BGP routers fails, the adja-

Manuscript received April 14, 2008; approved for publication by Min Young Chung, Division III Editor, March 12, 2009.

K. Xu is with the Arizona State University, Phoenix, AZ 85069, USA, email: kuai.xu@asu.edu.

J. Chandrashekar is with the Intel Research, Santa Clara, CA 95054, USA, email: jaideep.chandrashekar@intel.com.

Z.-L. Zhang is with the University of Minnesota, Minneapolis, MN 55455, USA, email: zhzhang@cs.umn.edu.

cent routers initiate routing events and send BGP updates to their neighbors. These updates indicate how "reachability" to certain destinations has changed. For example, if the failure caused a loss of reachability to a destination network, the router will generate a *withdrawal* message, listing the network prefixes that have become unreachable. On the other hand, if the failure simply causes a path change (or if the router learns of a previously unknown destination), then an *announcement* is generated—containing a set of network prefixes, and associated path attributes. A particularly relevant attribute is the AS PATH, which indicates both the origin AS for the prefix, as well as the sequence of ASes over which the route was propagated. Upon receiving a BGP update from a neighbor, a router might itself—after updating its own routing state—generate a *secondary* route update. Thus, by the mechanism just described, information about "events" propagates router by router through the network.

As a valuable service to the networking community, public "collection" sites such as Route-Views [13] and RIPE [14] maintain BGP peering sessions with a number of routers in various ISP's and log the received updates. For clarity, we will call the time ordered sequence of updates observed at a single vantage point as a BGP *update stream*, and these form the starting point for our methodology.

B. Principal Component Analysis

PCA (and its variant, factor analysis) is typically used to reduce the "dimensionality" of a data set and to uncover interrelated *latent* variables (or factors) in the original dataset. This is accomplished by projecting the original data onto a lower dimensional space in a manner that preserves most of the variance present in the original data. In the following, we present a brief algorithmic description of PCA, focusing on the relevant details (for a detailed discussion, see [15]).

Let $\mathbf{X} = [\mathbf{X}_1\mathbf{X}_2\cdots\mathbf{X}_p]^T$ be a $p \times t$ (observation) matrix of p variables on a time interval divided into t slots. In other words, for $i = 1, 2, \cdots, p$, the row vector $\mathbf{X}_i^T$ represents a time series of observations of a (observable) random variable, and X_{ij} is its observed value at time slot j. Given this matrix $\mathbf{X}$, PCA proceeds as follows:

1. The $(p \times p)$ covariance matrix $\mathbf{S} = \mathbf{X}\mathbf{X}^T$ is computed. S_{kl} is the covariance of the random variables $\mathbf{X}_k$ and $\mathbf{X}_l$.
2. Since $\mathbf{S}$ is square symmetric, all of its p eigenvalues are real. Let $\lambda_1, \lambda_2, \cdots, \lambda_p$ be the rank-ordered eigenvalues with corresponding eigenvectors $\alpha_1, \alpha_2, \cdots, \alpha_p$, i.e., $\mathbf{S}\alpha_i = \lambda_i\alpha_i$, and $\alpha_i^T\alpha_i = 1, 1 \leq i \leq p$. Note that the vectors $\{\alpha_j\}$ form an orthogonal basis for a p-dimensional space.
3. For $i = 1, 2, \cdots, p$, the ith principal component ($\mathbf{PC}_i$) is obtained by projecting the original data $\mathbf{X}$ onto the i dimension, i.e., $\mathbf{PC}_i = \alpha_i^T\mathbf{X}$.

   Since $\text{var}(\mathbf{PC}_i) = \text{var}(\alpha_i^T\mathbf{X}) = \alpha_i^T\mathbf{X}\mathbf{X}^T\alpha_i = \alpha_i^T\mathbf{S}\alpha_i = \lambda\alpha_i^T\alpha_i = \lambda_i$, we see that the variance captured by the ith principal component is exactly described by the ith largest eigenvalue. Also, α_1 is the the direction along which the original data has the largest variance and the fraction of variance captured is $\frac{|\lambda_1|}{\sum_i |\lambda_i|}$.

   Let $\mathbf{PC} = [\alpha_1\alpha_2\cdots\alpha_p]^T\mathbf{X}$. Then PCA transforms the space containing the "samples" of the p *observable* variables $\{\mathbf{X}_i\}$ into a new space of p principal components (*latent* variables) denoted as $\{\mathbf{PC}_i\}$, where the first variable $\mathbf{PC}_1$ contains the most variance inherent in the original data, and for $i = 2, \cdots, p$, the ith variable, $\mathbf{PC}_i$, contains most of the variance in the remaining data (after removing the contributions of the previous $i-1$ principal components).
4. The final step in PCA is to project the original dataset onto a (sub)space of reduced dimensionality to obtain an approximate representation that preserves most of the variance. To capture $\theta\%$ of the variance of the original dataset, we find the smallest m such that $\frac{\sum_{i=1}^{m}\lambda_i}{\sum_{j=1}^{p}\lambda_j} \geq \theta\%$. Then the projection is described as:

$$\hat{\mathbf{PC}} = [\alpha_1, \alpha_2, \cdots, \alpha_m]^T\mathbf{X}.$$

The utility of PCA lies in the fact that in most situations, $m \ll p$. In other words, the original data can be reduced (approximately) to a set of m dominant principal components (latent variables or factors) containing the most variance. Note that $\mathbf{PC}_i$ can be re-written as:

$$\mathbf{PC}_i = \alpha_i^T\mathbf{X} = [\alpha_{i1}\mathbf{X}_1 + \cdots + \alpha_{ip}\mathbf{X}_p]^T = [\sum_{j=1}^{p}\alpha_{ij}\mathbf{X}_j]^T. \quad (1)$$

Here, α_{ij}, $j = 1, \cdots, p$, is the coefficient (or *PC loading*) of $\mathbf{X}_j$ for $\mathbf{PC}_i$. It describes the contribution of $\mathbf{X}_j$ to the variance captured by the ith principal component. To state it differently, α_{ij} indicates the influence of the ith *latent* variable ($\mathbf{PC}_i$) on the variance of the *observable* variable $\mathbf{X}_j$. These properties of PCA are the key to our methodology, which uses PCA to exploit temporal correlation between BGP updates triggered by the same event.

III. METHODOLOGY

A. Constructing BGP Update Matrix

An overall schematic depiction of our methodology is shown in Fig. 1. In the first stage, we convert the update stream into an appropriate observation matrix called the (BGP) *update matrix*, denoted by $\mathbf{X}$ from a stream of BGP updates obtained during an observation interval at a single BGP vantage point. Let $Q = \{q_1, q_2, \cdots, q_l\}$ be the set of all prefixes for which at least one update (announcement or withdrawal) was observed in the interval, and let $A = \{a_1, a_2, \cdots, a_l\}$ be the set of corresponding *origin* ASes that own these prefixes.

To construct the update matrix $\mathbf{X}$, we first divide the observation interval into discrete time slots of size $\delta = 30$ seconds. This particular choice of δ is motivated by results presented in [16], [17], where it is shown that most updates for the same prefix arrive at multiples of approximately 30 seconds. In each time slot j, we calculate the number of distinct updates associated with an origin AS i, denoted by X_{ij}. In other words, each row is a time series of updates associated with each origin AS. For example, the ith row of the matrix $\mathbf{X}$, $(X_{i1}, X_{i2}, \cdots, X_{it})$, represents the number of updates for the origin AS i from the time T to $T+30t$ seconds. X_{ij} becomes 0 if there is no update for the origin AS i during the jth time slot.

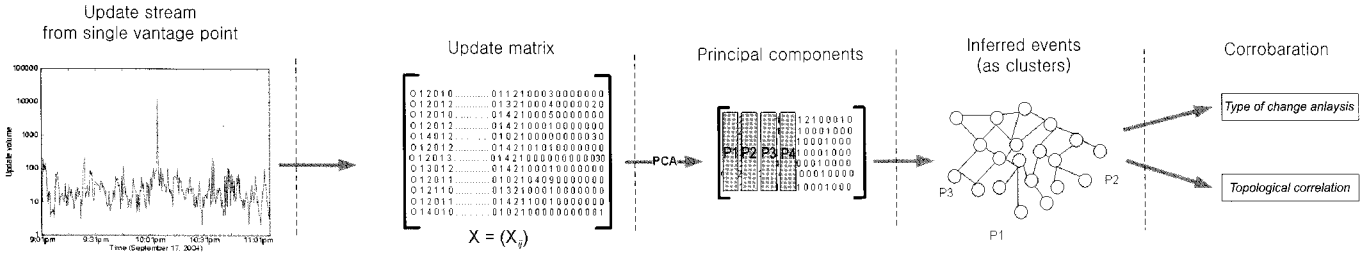


Fig. 1. Overview of our methodology.

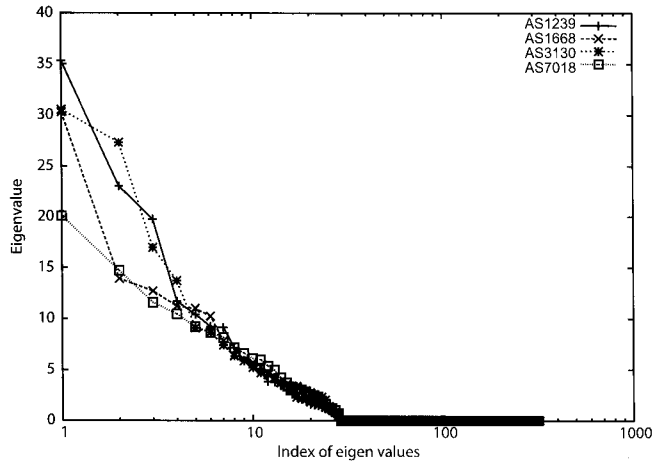


Fig. 2. Eigen value distributions of the update matrix.

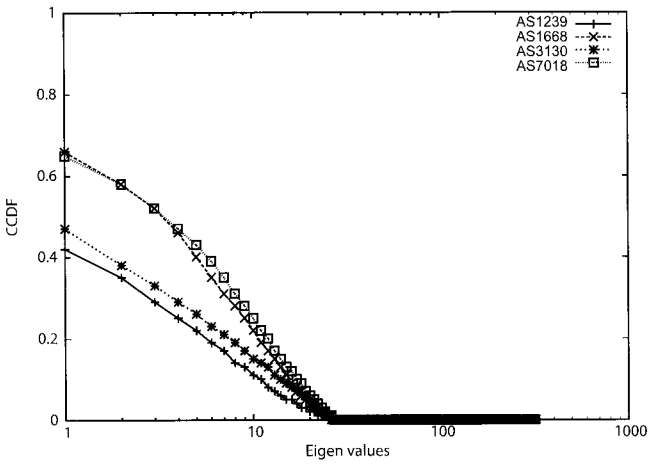


Fig. 3. Cumulative variance accounted for by top m PCs.

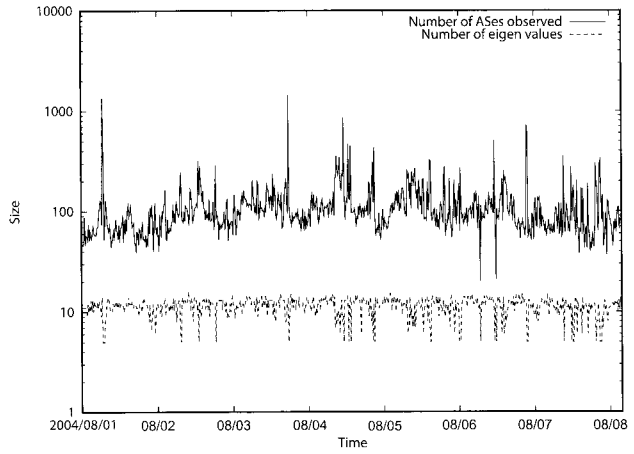


Fig. 4. Number of top m PCs accounting for most variations over a week.

To reduce the effects of AS size, we normalize each row of the update matrix into a standard form as follows. Let μ_i be the sample mean of the update signal associated with AS i, i.e., $\mu_i = \sum_j X_{ij}/t$ (t is the number of time slots in the interval), and σ_i^2 is the corresponding sample variance. Then the (normalized) update matrix is $\tilde{\mathbf{X}} = [\tilde{X}_{ij}]$, with $\tilde{X}_{ij} = (X_{ij} - \mu_i)/\sigma_i$. Now each row of $\tilde{\mathbf{X}}$, $\tilde{\mathbf{X}}_i^T = [X_{ij}, 1 \leq j \leq t]$, is a time series with a zero mean and unit variance, and represents the relative update "signal" strengths associated with AS i over the entire observation interval. For each AS i, the absolute value of $\tilde{X}_{ij}$ indicates how much the observed update signal at time slot j differs from the overall (mean) signal strength seen during the observation period.

B. Selecting Dominant Principal Components

The second stage of our methodology is to select the *dominant* principal components (PCs) that account for most of the variances in the update signals, based on their associated eigenvalues. The intuition here is that these *dominant* PCs statistically capture the underlying major network events that trigger the updates. Let $\{\lambda_i, 1 \leq i \leq p\}$ be the rank-ordered list of eigenvalues, the dominant PCs are selected based on the following two conditions: given a threshold θ, $0 < \theta \leq 1$, let m be the smallest integer such that

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{p} \lambda_i} \geq \theta$$

and $\lambda_m > 1$.

The first condition specifies the *desired cumulative variation* that the top m (*dominant*) PCs should account for. In practice, θ values in the range $[0.7, 0.9]$ are recommended [15], and in our own analysis, we choose $\theta = 0.8$. The second condition is referred to as Kaiser's criterion [18]. It signifies that each dominant PC should contain more variance than is associated with a *single* variable (recall that each row in the normalized update matrix has zero mean and unit variance). This test is important in our analysis as otherwise we can always find a value m that satisfies the first condition. By imposing Kaiser's criterion, we attempt to clearly separate "major" events that contribute large variance in the update streams from "minor" events.

In the following, we use an example to illustrate the process (and effect) of selecting the *dominant* PCs: We obtain (normalized) update matrices from BGP update streams of four distinct

vantage points, AS1239, AS1668, AS3130, and AS7018, corresponding to the same observation interval on August 2, 2004. Fig. 2 is a scree plot of the eigenvalues of the update matrices and Fig. 3 shows the corresponding cumulative variances associated with the rank-ordered eigenvalues. The latter figure clearly shows that a few (approx. 4–13) of the largest eigenvalues account for almost all the variance in the original data. Moreover, this number is an order of magnitude smaller than $p \approx 300$, which is the number of rows of $\mathbf{X}$. More importantly, this property does not depend upon the particular observation interval, as is shown in Fig. 4. Here, for each interval from a one-week long update stream (collected from a single vantage point, AS1239), we plot both the value of p (top curve) and m, the number of dominant PCs that account for more than 80% of the variance of the p original variables. It can be clearly seen that m is at most 15 in all the cases, while p is at least an order magnitude greater. These observations show PCA may be a useful tool to analyze BGP updates.

C. Extracting AS Clusters

Finally, we describe how each dominant PC is mapped to a set of (origin) ASes that are likely affected by the same underlying event. For ease of exposition, we refer to this set as an (origin) AS cluster. Extracting the AS cluster from each dominant PC will enable us to study the "common features" (e.g., spatial properties) shared by the ASes in the cluster, on dimensions other than the temporal one. In particular, the last stage of our methodology involves the use of topology and AS PATH information to locate similarities between ASes in the same cluster. Since the more detailed analyses are performed on a reduced set of statistically correlated updates, they can not only help validate and corroborate our methodology, but also yield potentially insightful hints on the possible root causes of the underlying events.

Recall from (1) that each dominant PC is a linear combination of the original observed variables (rows in the update matrix). For a dominant $\mathbf{PC}_i$, the coefficient (PC loading) α_{ij} reflects how much effect $\mathbf{PC}_i$ has on the variance of the variable $\bar{\mathbf{X}}_j$, namely, the (normalized) update signals from AS j. Let $\hat{a}_i = \max_{1 \le j \le n} \alpha_{ij}$ be the maximal value of the coefficients. Our intent is to select all ASes that contribute approximately the same loading. To do this, we select all coefficients $\alpha_{ij}, 1 \le j \le p$, such that $\alpha_{ij} \in [(1-\epsilon)\hat{a}_i, \hat{a}_i]$. The corresponding ASes are then grouped into an (origin) AS cluster associated with $\mathbf{PC}_i$. The underlying intuition is that the underlying "event" captured by $\mathbf{PC}_i$ is most likely to affect those variables (origin ASes) whose corresponding PC loadings are close to the maximal value; thus updates associated with these ASes are likely to be highly correlated. The parameter ϵ can be used to control the "tolerance" of correlation among ASes in the same cluster: Smaller values of ϵ will admit more ASes—with less strongly correlated updates—into the clusters. In our study, we have experimented with different values of ϵ in the range $[0.01, 0.10]$ and observed that the composition of the AS clusters does not change in a significant way across the range of values. For convenience, we use $\epsilon = 0.05$ for the remaining analysis.

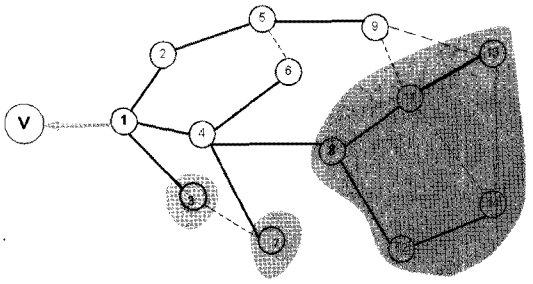


Fig. 5. Event sets in the simulated topologies. The bold lines indicate the actual path used by the vantage point. Dotted lines indicate links that are available, but not used.

IV. SIMULATIONS

A. Simulation Set-up

We simulate a number of topology families, including Waxman and Power-law topologies using the SSFNet simulator package [19]. To keep the description simple, we present the details of the simulations carried out on a single Power-law topology of size 400.

We simulate two distinct kinds of dynamic events—major and minor. For the former, we select nodes that have a high impact on the topology, cause them to fail at a particular time and then restore the node at a later time. For the minor events, we select nodes which have a very small impact on the topology and cause them to periodically fail and be restored. Note that here, a major event affects reachability to many ASes, while a minor event affects only one or at most a few ASes. Moreover, minor events are also periodic, generating persistent background update "noises," whereas major events have a large impact but last a smaller duration, triggering a burst of updates in a relatively short period of time. The insight for choosing major and minor events in this manner comes from the results presented in [4].

For each simulation run, we generate a set of 10 events, with a 60% of the events being minor and the remaining as major events. The arrival times for the events are generated from an exponential process with mean set to the convergence time of the topology.

B. Simulation Results

First, we study whether the clusters obtained using our method do in fact corresponded to the expected set of ASes. From the static topology, we determine the composition of AS clusters that we expect for each distinct event. Consider Fig. 5. Here when node 8 fails (or is repaired), we expect that the prefixes associated with nodes in the larger shaded region are affected, the updates for these will be seen at the vantage point. Note that only node 8 becomes unreachable–all other nodes can be reached via alternate paths. Similarly, when node 3 fails, the only node affected is itself (as shown in the figure). Thus, given the *path set* from the vantage point, we associate every event with an *expected set* of (origin) ASes that are affected by the event. The *expected sets* are then compared with the *inferred* AS clusters, obtained by applying our methodology to the updates collected at the vantage point during the simulation. For each cluster, we identify the "event node," i.e., the node affected by the failure (or repair). If an *inferred* AS cluster contains an

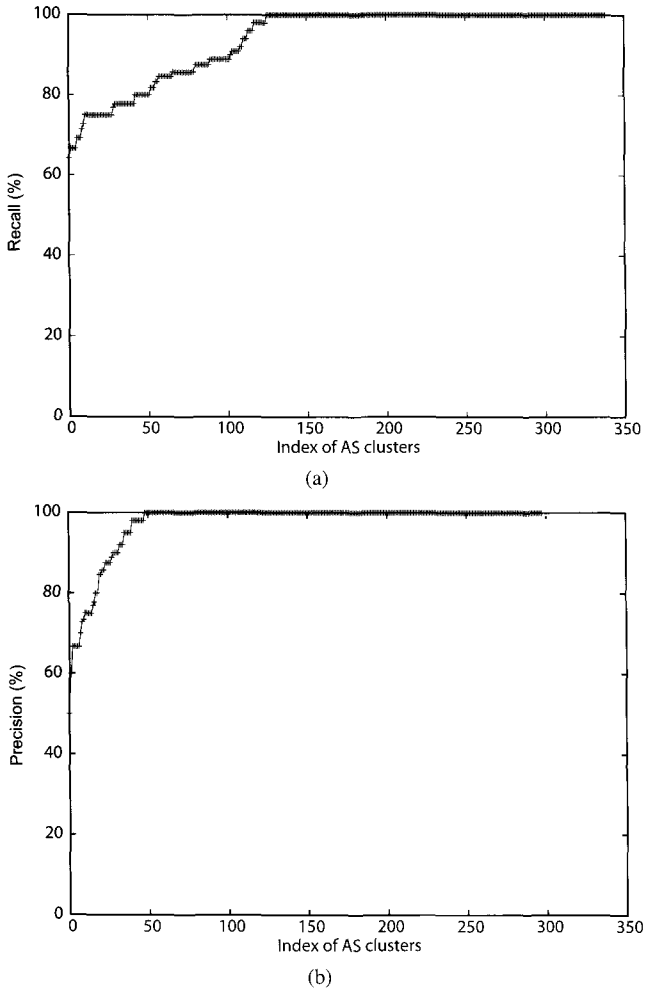


Fig. 6. CDF of *recall* and *precision*: (a) *recall* and (b) *precision*.

event node, we consider it as a *candidate* cluster.

Ideally, the *expected* sets should match up perfectly with the *inferred* AS clusters. In order to determine how good the matching is, we define two metrics, *recall* and *precision*, which we define as follows. Let S_E denote the *expected set* and S_I the *inferred* AS cluster. Finally, $S_M = S_E \cup S_I$ is the set of *matched* ASes that are common to both. Then, we define the *recall*, expressed as a percentage, as: $recall = \frac{|S_M|}{|S_E|} \times 100\%$ and the *precision* (also expressed as a percentage) as: $precision = \frac{|S_M|}{|S_I|} \times 100\%$.

We plot these two measures for *all* the "*inferred*" (major) events (there are 339 in all), over more than 100 simulations in Figs. 6(a) and 6(b). As shown in Fig. 6(a), the *recall* for most *inferred* events is over 80%, while the average value is 93.1%. This indicates that in almost all the cases, our methodology places most of the *affected* ASes in the *inferred* AS clusters. Similarly, as shown in Fig. 6(b), the precision is over 90% for all but a few events, indicating that the *inferred* clusters capture those elements that are in the appropriate *expected set* with very low noise. These results show that our methodology captures the (simulated) events with high accuracy.

Next, we study how effective our methodology is in separating ASes/updates associated with distinct events, even when they are occurring concurrently? To address this question, we set up the simulations such that multiple (major and minor) events often occur close together, triggering updates that are mixed at the vantage point. In order to quantify "concurrent" events, we count the number of *overlapping* events corresponding to each *inferred* event. Note that each event can be associated with a set of timestamps. For example, event i is associated with the set of timestamps $\tau_i = \{t_{i,1}, t_{i,2}, \cdots, t_{i,m}\}$, where each timestamp $t_{i,k}$ is the time at which the kth dynamic occurs (which could be a failure or a repair event). For major events, we have $k \leq 2$. Given a major event i and any other event j, we say that i and j are overlapping events if they contain timestamps that are β seconds apart. Formally, event j overlaps with event i if and only if

$$\exists t_{i,k} \in \tau_i, \exists t_{j,l} \in \tau_j \text{ such that } |t_{i,k} - t_{j,l}| \leq \beta.$$

Thus, events overlap if they trigger updates within 60 seconds (30 seconds) of each other.

For each major event that is inferred, we count the number of overlapping events. The results show that, on average, there are about 2 other events that overlap with each major event. Most of the *major* events overlap with one or two other events, and there are a few events that overlap with more than 3 events. However, in spite of multiple overlapping events which generate interleaved updates, our methodology has very high recall and precision. This shows that our methodology is indeed very effective in separating updates triggered by distinct events that occur close together.

V. REAL BGP UPDATE STREAMS

We now present the results of applying our methodology upon real BGP data collected from Route-Views. In particular, we analyze data collected in Aug. 2004 and Sept. 2004. Our first finding is that major events occur relatively often. The median number of events in each (approx. 15 min long) interval is 12 (for both the datasets that we have analyzed). These observations indicate that routing events occur frequently and often close together, triggering BGP updates that are likely to be interleaved.

Next, we study the duration of the "*inferred*" events, defined as follows. For each AS cluster, we reconstitute the actual updates for every AS in the cluster; in other words, for each AS in a given cluster, we reproduce the original update stream by filtering the appropriate prefixes. Then the duration of each prefix-specific event is the the time elapsed between the first and last updates. The event duration is the largest prefix-specific duration among all those obtained from the same AS cluster. As observed in the results, 72% of all the events last less than 180 seconds (3 minutes). In particular, there are a few events that last close to 900 seconds. We believe that these correspond to the "persistently flapping" events, described in [4], which last for long periods of time. However, since we divide the update stream into disjoint intervals, such events are independently inferred in each of the intervals.

The "impact" of the *inferred* events can be defined either in terms of the number of ASes affected or in terms of the number of affected network prefixes. We find that 90% of the *inferred* events contain fewer than 11 ASes, and in 90% of the events, less than 52 network prefixes are impacted, i.e., there is

a change in reachability. At the same time, there are also a number of events which affect hundreds (even thousands) of ASes and network prefixes in some observation intervals. We expect that these *inferred* events can be traced to large scale routing events.

To validate and corroborate that ASes in the same cluster obtained via our PCA analysis are plausibly affected by the same actual event, we introduce three metrics as a measure of "common feature" that are shared by the ASes in the cluster: *Dominant change type, dominant provider, and dominant country*

The metric of *dominant change type* is developed based on the "type of change" classification discussed in [4]. For a given cluster, the *dominant change type* is the class associated with the most prefixes in the cluster. For example, if some cluster C is associated with $k_1 + k_2$ prefixes, such that k_1 is of type re-route, k_2 is to of type prefix-up, with $k_1 > k_2$, then the *dominant change type* is re-route. Also, k_1 is referred to as the size of the *dominant change type* set. The dominant change type accounts for more than 80% in over 80% of the events. The results indicate that in the case of most large events, i.e., associated with large AS clusters, almost all the prefixes are affected in the same way. Thus, it is plausible to believe that the ASes (and prefixes) in the cluster are affected by the same event.

Given an AS cluster, we identify the *dominant provider* as follows: For each AS in the cluster, we first obtain the AS Path from the vantage point to the AS prior to the event, in other words the *stable path(s)* for the prefixes in the cluster. By treating these *stable paths* as a directed set of edges, we construct a tree-like subgraph with the vantage point as the root. With each node x in this subgraph, we associate a value $p(x)$, which is simply the number of downstream customers that belong to this cluster [20]. Note that $p(x)$ is exactly the number of ASes that would be affected by an event at x. Finally, the *dominant provider* is the node $\hat{x}$: $p(\hat{x}) \geq p(x)$, i.e., the node with the largest value. For clarity, we describe $p(\hat{x})$ as the *dominant provider contribution*. In general, the *dominant provider contribution* accounts for more than 80% of the size of the AS cluster in over 88% of the events. Thus, the AS clusters identified by our methodology are likely affected by the same network event.

Given an AS cluster, we could easily compute the percentage of ASes in each country as its contribution of the cluster. In addition, we sort these countries based on the percertages in a non-increaing manner. As a result, the first (dominant) country always has the largest contribution to the cluster. As observed in the real BGP data, most ASes within one cluster are from the same countries. These observations are consistent with the above intuition that region routing events often trigger routing updates of ASes in the same region. On the other hand, there are a few cases, especailly, those large AS clusters, in which the first country explains only part of ASes in the cluster. Such clusters are likely caused by the major routing events which have a global impact.

To summarize, in most of the large events (i.e., with large AS cluster size), the ASes and associated prefixes contained in the cluster seem to share strong common features that may be traced to the same event. Thus, based on these observations, we may plausibly believe that the AS clusters obtained through our methodology are likely to have their genesis in *actual, distinct* events [21].

VI. RELATED WORK

Understanding BGP dynamics and their underlying "root causes" is an extremely challenging problem due to the inherent complexity of inter-domain routing. Recently, a number of efforts have tried to address this problem. In all of these efforts, the goal is to infer the approximate location of routing instabilities by analyzing BGP updates collected at multiple vantage points along three independent dimensions—time, prefix, and vantage point. However, as discussed in [12], there are several pitfalls associated with inferring events based only upon BGP update data. In our work, rather than attempt the harder problem of identifying the location of routing events, we use a statistical approach to separate updates triggered by distinct events. In particular, the results that we present can serve to "inform" the traditional approaches to performing root cause analysis.

In [22], Andersen et al., use a clustering technique upon BGP updates collected over a long interval to identify "hidden" topological relationships between network prefixes. The intuition is that prefixes that are updated together over a very long period, then it is likely that they are located close to each other. While our work also shares this underlying intuition, our objectives and methodology are very different. In particular, we are looking at correlations over a shorter time with the intent of inferring network events.

VII. CONCLUSIONS

In this paper, we have proposed a novel methodology for identifying and separate BGP updates associated with major events. The methodology is based on PCA, a well-known multivariate data analysis technique, which enables us to exploit the temporal correlations in the update streams to extract clusters of origin ASes whose prefixes are likely affected by the same network events. Subsequently, we perform spatial correlation and "type-of-change" analysis on the extracted AS clusters and their associated updates to further validate and corroborate our findings. Through extensive simulations and evaluations using real BGP update streams, we find that in most cases, ASes in a cluster exhibit the same type of routing changes and/or are well correlated spatially (in a topological sense). We believe that our methodology can potentially help characterize the nature of the BGP update streams and narrow down the problem space for root cause analysis and trouble-shooting.

REFERENCES

[1] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," RFC 1771, Mar., 1995.

[2] T. Griffin, "What is the sound of one route flapping?" Netw. Modeling and Simulation Summer Workshop, 2002.

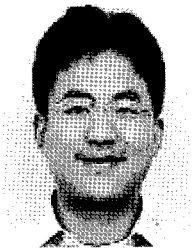
[3] D. Chang, R. Govindan, and J. Heidemann, "The temporal and topological characteristics of BGP path changes," in *Proc. ICNP*, 2003.

[4] M. Caesar, L. Subramanian, and R. Katz, "Root cause analysis of Internet routing dynamics," U.C. Berkeley, Tech. Rep. UCB/CSD-04-1302, Nov. 2003.

[5] A. Feldmann, O. Maennel, Z. Mao, A. Berger, and B. Maggs, "Locating Internet routing instabilities," in *Proc. ACM SIGCOMM*, 2004.

[6] M. Lad, D. Massey, and L. Zhang, "Link-rank: A graphical tool for capturing bgp routing dynamics," in *Proc. IEEE/IPIF NOMS*, Apr. 2004.

[7] J. Rexford, J. Wu, Z. M. Mao, and J. Wang, "Finding a needle in a haystack: Pinpointing significant bgp routing changes in an ip network," in *Proc. NSDI*, 2005.
[8] D. Massey, M. Lad, R. Oliveira, and L. Zhang, "Inferring the origin of routing changes using link weights," in *Proc. Int. Conf. Netw. Protocols*, 2007.
[9] J. Gottlieb, L. Wang, M. Saranu, and D. Pei, "Understanding bgp session failures in a large isp," in *Proc. INFOCOM*, 2007.
[10] Z. M. Mao, Y. Zhang, and M. Zhang, "Effective diagnosis of routing disruptions from end systems," in *Proc. NSDI*, 2008.
[11] K. Xu, J. Chandrashekar, and Z.-L. Zhang, "A first step towards understanding inter-domain routing," in *Proc. ACM SIGCOMM Workshop on Mining Netw. Data*, Philadelphia, 2005.
[12] R. Teixeira and J. Rexford, "A measurement framework for pin-pointing routing changes," in *Proc. ACM SIGCOMM Netw. Troubleshooting Workshop*, 2004.
[13] University of Oregon. Routeviews archive project. [Online]. Available: http://archive.routeviews.org/
[14] RIPE. Routing information service raw data. [Online]. Available: http://data.ris.ripe.net/
[15] I. T. Jolliffe, *Principal Component Analysis, 2nd ed.* Spinger Series in Statistics, 2002.
[16] M. Mao, R. Bush, T. Griffin, and M. Roughan, "BGP beacons," in *Proc. Internet Meas. Conf.*, 2003.
[17] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Trans. Netw.*, 2001.
[18] H. F. Kaiser, "The application of electronic computers to factor analysis," *Educational and Psychological Meas.*, 1960.
[19] SSFNET. Scalable simulation framework. [Online]. Available: http://www.ssfnet.org
[20] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM*, 2002.
[21] K. Xu, J. Chandrashekar, and Z.-L. Zhang, "Inferring major events from BGP update streams," University of Minnesota, Dept. of Computer Science, Tech. Rep. 04-043, 2004.
[22] D. Andersen, N. Feamster, S. Bauer, and H. Balakrishnan, "Topology inference from BGP routing dynamics," in *Proc. Internet Meas. Workshop*, Nov., 2002.



Kuai Xu is currently an Assistant Professor at Arizona State University. He received his Ph.D. degree in Computer Science from the University of Minnesota in 2006, and his B.S. and M.S. degrees in Computer Science from Peking University, China, in 1998 and 2001. His research interests include network security and cloud computing. He is a Member of ACM and IEEE.

Jaideep Chandrashekar received a B.E. degree from Bangalore University, India, in 1997 and a Ph.D. from the University of Minnesota in December 2005. He is currently with Intel Research in Santa Clara, CA. His research interests include computer networks and distributed systems, especially Internet technologies, network routing, and computer security. He is a Member of ACM and IEEE.

Zhi-Li Zhang received the B.S. degree in Computer Science from Nanjing University, China, in 1986 and, his M.S. and Ph.D. degrees in Computer Science from the University of Massachusetts in 1992 and 1997. In 1997, he joined the Computer Science and Engineering Faculty at the University of Minnesota, where he is currently a Professor. His research interests include computer communication and networks. He is co-recipient of an ACM SIGMETRICS Best Paper Award and an IEEE International Conference on Network Protocols (ICNP) Best Paper Award. He is a Member of IEEE, ACM, and INFORMS Telecommunication Section.