Inferring Major Events from BGP Update Streams
Technical Report 04-043

Kuai Xu, Jaideep Chandrashekar, Zhi-Li Zhang
Department of Computer Science & Engineering
University of Minnesota at Twin Cities
kxu, jaideepc, zhzhang@cs.umn.edu

Abstract

BGP updates are triggered by a variety of events such as link failures, session resets, router crashes, policy or configuration changes. Making sense of BGP update streams and inferring their underlying causes is important in trouble-shooting BGP and improving its performance. In this paper we propose a novel methodology to identify BGP updates associated with major events– affecting network reachability to multiple ASes– and separating 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 (PCA), which enables us to transform and reduce the BGP updates into different AS clusters that are likely affected by distinct major events. We also perform "spatial correlation" and "type-of-change" analysis based on AS PATH attributes to further validate and corroborate our findings. We demonstrate the accuracy and effectiveness of our methodology through simulations, and subsequently apply it to real BGP data. In addition, we corroborate our approach by analyzing updates corresponding to periods in which well-known routing events took place.

I. Introduction

BGP [1], the *de facto* Internet inter-domain routing protocol is an *incremental* path vector protocol: routing updates (announcements and withdrawals) are generated *only* in response to *network events*, such as link or router failures (or repair), session resets, policy change, misconfiguration, etc. A BGP router may receive thousands of BGP updates on a daily basis, reflecting activity from all over the Internet. Given the critical nature of the Internet routing infrastructure, understanding BGP routing dynamics and the underlying "root causes" is crucial, but at the same time, very challenging.

In the recent past, several efforts have been directed at the "root cause analysis" of BGP updates [2], [3], [4], [5], [6], 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 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/or hide the actual events [7]. Second, different events may trigger similar types of updates, making it hard to distinguish the events based solely 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 (perhaps) simpler question that is yet fundamental 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.

Problem Motivation and Our Methodology. BGP updates are triggered by a variety of events. Some events may affect prefixes originated from only a few (*origin*) ASes, generating only a small number of BGP updates. For example, a network failure in a *stub* AS[1] will trigger a route withdraw from the stub AS. A policy change relating to a few customer prefixes at a tier-1 or tier-2 ISP will also generate only a few BGP updates. On the other hand, a link or router failure within a tier-1 AS or a BGP session reset between two tier-1/tier-2 ASes may trigger a burst of BGP updates, since the event not only affects relevant network prefixes originated by the said AS(es) but also those from their customers. Since network events may occur concurrently or at similar time, BGP updates from these events are likely to be interleaved as they propagate across the Internet. Hence, a burst of BGP updates observed by a distant BGP vantage point can be caused by a number of unrelated events occurring on the Internet. On the flip side, a burst of BGP updates *which may seem unrelated* –looking at the associated AS PATHs – may actually be caused by the same underlying event. For example, a failed switch at an Internet Exchange Point may affect multiple ASes that do not have direct connections, generating a burst of BGP updates whose AS PATHs may not intersect. As another example, a major fiber cut (e.g., the Baltimore tunnel fire in 2001) may affect many ASes whose backbone networks go through the same fiber track, causing a flurry of "unrelated" BGP updates. These examples, together with the routing policy complications illustrated in [7], not only highlight the potential pitfalls and limitation of traditional BGP root cause analysis techniques, but also point to the need for *techniques to identify and separate BGP updates associated with various events*—the key question we attempt to address in this paper.

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* (or PCA), a well-known statistical method for multivariate data analysis [9]. 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).

Contributions. Our contributions, as presented 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. Applying our methodology to two months of BGP updates, we show that in general, there are between 2 to 15 *major* events occurring every 15 minutes.
- The results presented in this paper can serve to *inform* and *guide* algorithms used to perform BGP root cause analysis and trouble-shooting. For example, by applying our analysis, the size of the "candidate sets" can be reduced, leading to a smaller set of event locations and better identification of plausible causes [5]. We present case studies to illustrate the utility of our methodology in such a setting.

The remainder of this paper is organized as follows. Section II briefly discusses the BGP operations and gives a high level overview of PCA. In Section III, we describe in detail how we apply our methodology upon BGP update streams to separate and identify updates associated with major events. 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. Section VI presents case studies using known BGP routing problems. We discuss related work in

[1]A stub AS is an AS that does not carry transit traffic, i.e., has no customers [8].

Section VII. Finally, Section VIII concludes the paper.

II. A Quick Primer on BGP and PCA

In this section, we first briefly discuss the operation in BGP, focusing on details relevant to this paper. In particular, we discuss exactly why the "root cause analysis" problem is challenging. Subsequently, we provide a very high level overview of Principal Component Analysis (PCA), as a precursor to a detailed description of our methodology in the following section.

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*.[2] Suppose a session between a pair of BGP routers fails, the adjacent 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 [10] and RIPE [11] maintain BGP peering sessions with a number of routers in various ISP's and log the received updates. Each of these routers acts as a *vantage point* into the Internet. Thus, in a sense, the updates that are logged reflect network events that have occurred somewhere. 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 [9]).

Let $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_p]^T$ be a $p \times t$ (observation) matrix of p variables on a time interval divided into t slots[3]. In other words, for $i = 1, 2, \ldots, p$, 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 $\lambda_1, \lambda_2, \ldots, \lambda_p$ be the rank-ordered eigenvalues with corresponding eigenvectors $\alpha_1, \alpha_2, \ldots, \alpha_p$, 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 $i = 1, 2, \ldots, p$, the i'th principal component ($\mathbf{PC}_i$) is obtained by projecting the original data X onto the i dimension, i.e., $\mathbf{PC}_i = \alpha_i^T \mathbf{X}$.
   Since $var(\mathbf{PC}_i) = var(\alpha_i^T \mathbf{X}) = \alpha_i^T \mathbf{X} \cdot \mathbf{X}^T \alpha_i = \alpha_i^T \mathbf{S} \alpha_i = \lambda \alpha_i^T \alpha_i = \lambda_i$, we see that the variance captured by

[2]For clarity, we distinguish between network events, such as link failure (or repair), session resets, policy changes, etc., and *routing events* by which we take to mean the generation of BGP routing messages by a router.

[3]We use X^T to denote the matrix transpose of X.

the i'th principal component is exactly described by the i'th 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 $\mathbf{PC} = [\alpha_1, \alpha_2, \ldots, \alpha_p]^T \mathbf{X}$. 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 $i = 2, \ldots, p$, 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:

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

The utility of PCA lies in the fact that in most situations, $m << p$. 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 PC_i can be re-written as:

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

Here α_{ij}, $j = 1, \ldots, p$, is the coefficient (or *PC loading*) of $\mathbf{X}_j$ for $\mathbf{PC}_i$. It describes the contribution of X_j to the variance captured by the ith principal component. To state it differently, α_{ij} indicates the influence of the i'th *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

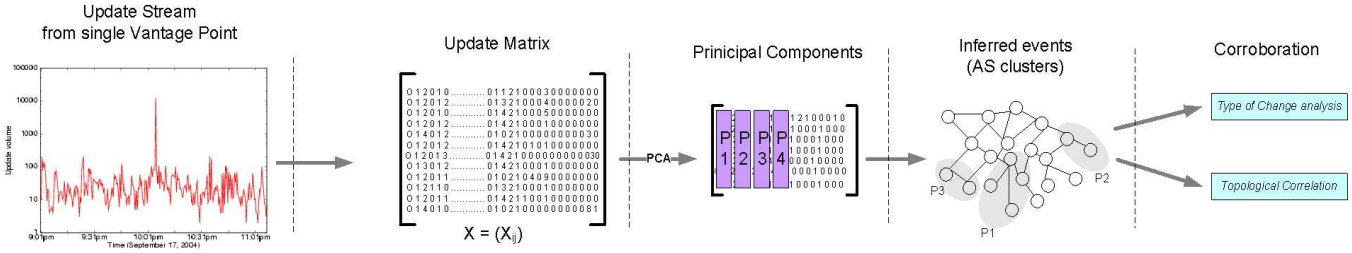


Fig. 1. Overview of our methodology

In this section, we describe how we apply our methodology upon BGP update streams to identify and separate updates associated with major events. An overall schematic depiction of our methodology is shown in Fig. 1. In the first stage, we convert a BGP update stream (from a single vantage point) into an *update matrix*, where each row represents a sequence of (normalized) "update signals" associated with each origin AS. Then we apply PCA on the *update matrix* to obtain a set of (dominant) principal components (or "inferred events") that account for most of the variance in the update signals. Subsequently, for each principal component, we reconstruct the set of origin ASes whose reachability (network prefixes) is affected by the "inferred event". In a sense, PCA transforms the BGP *update space*, represented by the update matrix $\mathbf{X}$, into an (underlying) *event space*; points in this space trigger the observed updates. The subspace spanned by the *dominant* principal components contains the "major events" that contribute a large fraction of the variance in the (observed) BGP update stream.

Finally, we validate and corroborate that BGP updates associated with an AS cluster thus obtained are plausibly caused by the same event. In order to do this, we identify topological and "type of change" correlations among the ASes in the cluster. In the following, we describe the first three stages, and the last stage is presented in section V.

A. Constructing BGP Update Matrix

Given a stream of BGP updates obtained during an observation interval at a single BGP vantage point, our first task is to convert the update stream into an appropriate observation matrix called the (BGP) *update matrix*, denoted by $\mathbf{X}$. Let $Q = \{q_1, q_2, \ldots, q_l\}$ be the set of *all* prefixes for which at least one update[4] (announcement or withdrawal) was observed in the interval, and let $A = \{a_1, a_2, \ldots, a_n\}$ be the set of corresponding *origin* ASes that own these prefixes.

In the paper we choose to consider (origin) ASes instead of prefixes as the (observable) variables to generate the update matrix; in other words, each row is a time series of updates associated with each origin AS (instead of a prefix). This is based on two important considerations: First, in this paper we are primarily interested in how (major) network events affect various ASes– we are less interested in individual prefixes– in order to obtain statistical analysis to assist in identifying and locating such events. In other words, the "granularity" of our update correlation analysis is at the AS-level instead of prefix-level.[5] Second, the number of prefixes are far larger than the number of ASes, resulting in an update matrix with potentially large dimension; on the other hand, using an AS-level update matrix significantly reduces the dimensionality (and computation time) for the PCA algorithm.

We divide the observation interval into discrete time slots of size $\delta = 30$ seconds. This particular choice of δ is motivated by results presented in [12], where it is shown that most updates for the same prefix arrive at multiples of approximately 30 seconds[6]. In each time slot j, we calculate the number of *distinct* updates associated with an *origin* AS i, denoted by X_{ij}. Since ASes are of greatly different sizes (i.e., originate different number of prefixes), a "minor" event, say, a link failure in a large stub AS, may result in more updates being generated than if the same event affected a smaller stub AS. As stated earlier, our objective is to identify *major events*—events that affect network reachability to a plurality of ASes (e.g., a link failure in a tier-1 AS that affects both itself as well as its AS customers)— and separate their associated updates from those triggered by *minor events*.

To mask 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.

Before we move on to describe the next stage, we comment on the choice of observation intervals. In our analysis, we partition the (continuous) update stream from a particular BGP vantage point into observation intervals of *approximately* 15 minutes, where the interval boundaries are adjusted such that large events (i.e., associated with a large number of updates) are completely contained in one or the other interval. The choice of 15 minutes is mostly due to convenience[7] and computational efficiency (larger intervals introduce more rows into the update matrix). We have performed the same analysis over longer periods (multiple of 15 minutes), which produced essentially the same results as if the analyses were performed on individual 15-minute intervals. This is likely because the effects of most network events last fewer than 15 minutes [13].

[4] A single BGP announcement or withdrawal can contain many prefixes. We preprocess the data and separate these into *update atoms*, each associated with a distinct prefix.

[5] We make the implicit assumption that only one major event (with distinct temporal characteristics) affects a particular AS at a given time. Although this assumption can be violated in theory, such events seem rare. We have done some initial analysis at the prefix-level, and found that the temporal correlations among prefixes within an AS are preserved and carry over to the AS-level, indicating that PCA at the AS-level is in general sufficient to capture such temporal correlations.

[6] This is mostly due to the fact that most several router vendors use a default of 30 seconds for the MRAI (minimum route advertise interval) timer.

[7] Route-Views archives updates from each BGP monitor every 15 minutes.

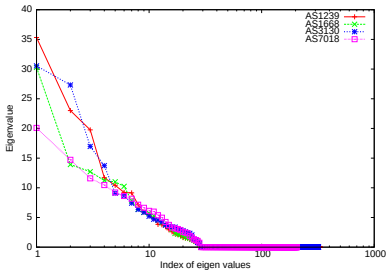


Fig. 2. Eigen value distributions of the update matrix

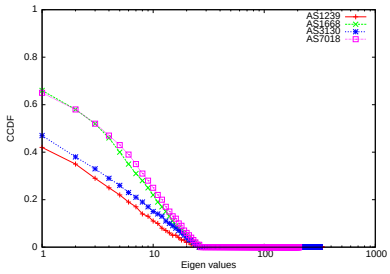


Fig. 3. Cumulative variance accounted for by top m PCs.

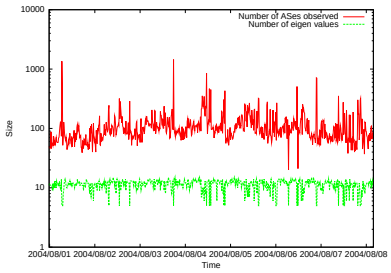


Fig. 4. Number of top m PCs accounting for most variations over a week.

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* principal components 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 [9], and in our own analysis, we choose $\theta = 0.8$. The second condition is referred to as Kaiser's Criterion [14]. 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"[8] 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, AS 1239), 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

[8]We can control and adjust "major-ness" of events by varying θ on cumulative variation as well as by using a stronger condition in choosing eigenvalues, e.g., $\lambda_m > a$ for some integer $a > 1$.

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 eq. (1) that each dominant PC is a linear combination of the original observed variables (rows in the update matrix). For a dominant PC_i, the coefficient (PC loading) α_{ij} reflects how much effect PC_i has on the variance of the variable $\tilde{\mathbf{X}}_j$, namely, the (normalized) update signals from AS j. Let $\hat{a}_i = \max_{1 \leq j \leq n} \alpha_{ij}$ 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 $\alpha_{ij}, 1 \leq j \leq p$, such that $\alpha_{ij} \in [(1-\epsilon) \times \hat{a}_i, \hat{a}_i]$. The corresponding ASes are then grouped into an (*origin*) AS cluster associated with PC_i. The underlying intuition is that the underlying "event" captured by 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.

In the next section, we apply our methodology to "simulated" update streams, in order to verify its effectiveness in identifying *major events*.

IV. Simulations

In the previous section we have described how the PCA methodology can be used to extract (origin) AS clusters that are likely associated with major events. In this section, we verify and demonstrate the *effectiveness* of our methodology in a *controlled* setting using simulations. Specifically, we attempt to answer the following two questions: i) *Are the* AS clusters *obtained via PCA likely to correspond to actual events? ii) Can the effects of distinct events be separated?* To answer these questions, we simulate events on synthetically generated AS topologies using the SSFNet simulator package [15], which contains support for BGP.

A. Simulation Set-up

We simulate a number of topology families, including Waxman and Power-law topologies.[9] To keep the description simple, we present the details of the simulations carried out on a single Power-law topology of size 400. For simplicity, we assume that each AS contains a single router and we use a simple route selection policy: BGP routers always prefer shorter AS paths (with the routerID acting as the tie breaker). To the simulated topology, we attach an additional node to the highest degree AS (i.e., with a peering session set up with the BGP router), to act as a vantage point. The vantage point was configured to dump all the updates received from its peer to a file in binary format (similar to what is done in Route-Views).

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].

[9] All the topologies were generated using BRITE topology generator, using default parameters.

In the following, we briefly describe how we select nodes for each kind of event. To begin with, we run the simulation without any dynamic events and determine the *path set* of the vantage point, i.e., the set of paths from the vantage point to each destination AS. Subsequently, with each node, we assign a *transit weight*, which is defined as the number of paths in the *path set* that transit this particular node. For example, in Fig. 5, the weight associated with node 3 is 1—there is a single path containing node 3, i.e., $(V, 1, 3)$ in the *path set*. Similarly, the weight of node 11 is 2 (the corresponding paths are $(V, 1, 4, 8, 11)$, $(V, 1, 4, 8, 11, 13)$). Intuitively, to simulate major events, we select nodes with *high* transit weight, since a failure of the node will affect *all* the destinations reachable through it. The exact node selection procedure is as follows: sort the nodes in non-decreasing order of *transit weight*; then for a minor event, select a node from the beginning of this set. On the other hand, for a major event, select nodes from the tail of this ordered set. The specific ranges from which major and minor events are generated are shown in Table I.

Event Type	Select from range
major	$[0.9, 1.0]$
minor	$[0.2, 0.7]$

TABLE I

RANGES FOR EVENT SELECTION.

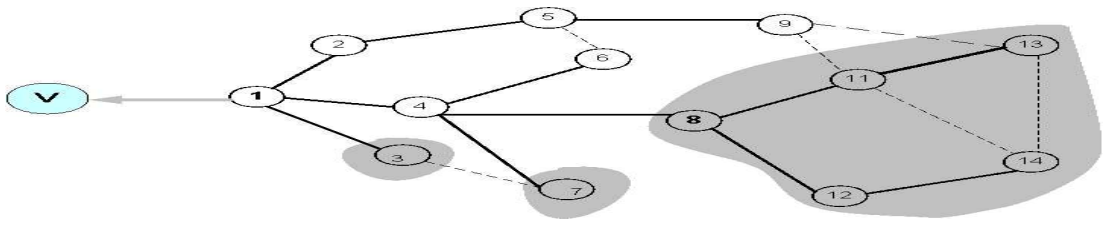


Fig. 5. Event sets in the simulated topologies. The bold lines indicate the actual path used by the vantage point. Dotted lines indicate links that are available, but not used.

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[10]. The arrival times for the events are generated from an exponential process with mean set to the convergence time of the topology. In the case of persistent (minor) events, we use a fixed duration between consecutive dynamics, but the actual number of cycles (failure followed by repair) is a random integer between 1 and 8.

We perform more than a hundred simulation runs on this particular topology and in each case, computed the *AS clusters* as described in the previous section. In this simulation study, departing slightly from our earlier description, we consider the entire simulation period as a single observation interval (instead of dividing the update stream into intervals of approximately 15 minutes, as described in the previous section): since there are relatively few ASes, it does not considerably worsen the computation time of the PCA algorithm.

B. Simulation Results

First, we present some results answering the first question posed in the beginning of the section, namely, 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

[10]The exact number of each type is determined by a random process.

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 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:

$$precision = \frac{|S_M|}{|S_I|} \times 100\%.$$

Intuitively, the *recall* captures how well the inferred set matches the expected set. Note that perfect (100%) *recall* simply means that *inferred cluster* contains *all* the elements of the *expected set* (but it can possibly contain other elements); the lower the recall is, fewer elements in the expected set are captured by the inferred set. On the other hand, *precision* captures how the inferred set matches against the expected set. Perfect (100%) *precision* indicates that only elements of the expected set are present in the inferred set (but some elements in the expected set may be missing in the observed set); the lower the precision, the more noise in the inferred set. To illustrate the difference between these two metrics, suppose a single event triggers BGP updates from 9 ASes, which constitute the expected set. By applying our methodology, we obtain an AS cluster of size 10; 9 of which are also contained in the *expected set*. Thus, in this example, the recall is 100% (since all the items in the *expected set* are matched), while the *precision* is 90% (since the observed cluster contains one item that is not in the expected set). Together the two metrics provide a measure how *accurate* our methodology is in capturing ASes affected by an event.

We plot these two measures for *all* the "inferred" (major) events (there are 339 in all), over more than 100 simulations in Figs. 6 and 7. As shown in Fig. 6, 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. 7, 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*.

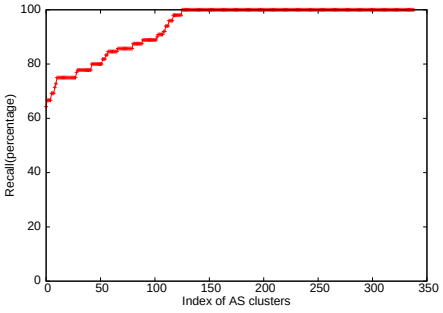


Fig. 6. CDF of *recall* plotted for 339 events obtained from 112 simulation trials.

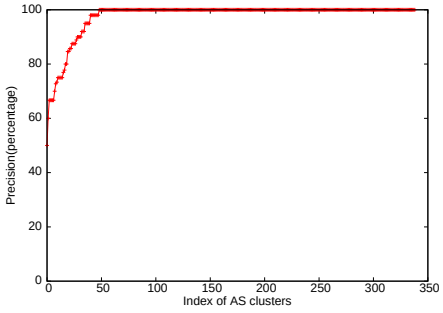


Fig. 7. CDF of *precision* plotted for 339 events from 112 simulation trials.

We now turn our attention to the second question posed earlier, namely, how effective is 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 $\tau_i = \{t_{i,1}, t_{i,2}, \ldots, t_{i,m}\}$, where each timestamp $t_{i,k}$ is the time at which the k'th dynamic occurs (which could be a failure or a repair event). For *major* events, we have $k <= 2$. 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. Table II shows the number of overlapping events for all major events in our simulations with $\beta = 30, 60$.

Overlapping events	0	1	2	3	4	5	6
# major events ($\beta = 60sec$)	0	102	125	81	23	4	4
# major events ($\beta = 30sec$)	15	98	120	78	21	4	3

TABLE II

A BREAKDOWN OF OVERLAPPING EVENTS OCCURRING CLOSE IN TIME WITH MAJOR EVENTS IN SIMULATIONS.

The results in the table 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 Sep. 2004, which we denote AUG and SEP respectively. For conciseness, we restrict our attention to two particular vantage points, i.e., AS 1239 (Sprint) and AS 7018 (AT&T).

First, we present some general observations about the (major) "events" *inferred* by our methodology, shedding light on the extent of BGP routing dynamics prevalent on the Internet. We then attempt to validate that the inferred events are *plausible*, i.e., it is reasonable that they correspond to actual Internet events. While in general it is hard to accurately determine this – the broader root cause analysis problem is very hard and statistics about Internet events are not available, we present three simple metrics that increases our confidence that this is indeed the case.

A. General Observations of BGP Events

In this part, we attempt to answer the following questions: *How often do events, particularly* major *events, occur? How long do they typically last? what is the impact of different events?* The statistics and observations presented are key contributions of this paper. To the best of our knowledge, ours is the first effort to try to quantify the BGP routing dynamics occurring on the Internet.

In Fig. 8, we show how often *major events* occur on the Internet. The figure plots the cumulative distribution (CDF) of the number of "inferred" events in each observation interval for the AUG dataset (for both vantage points). The figure clearly indicates 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). Interestingly, there are no "quiet" intervals (i.e., intervals in which no events are inferred) in the entire month. The maximum number of events inferred in any interval over the entire month is 15 (and 16) for AS 1239 (and AS 7018). These observations indicate that routing events occur frequently and often close together, triggering BGP updates that are likely to be interleaved. The statistics are similar in the SEP dataset.

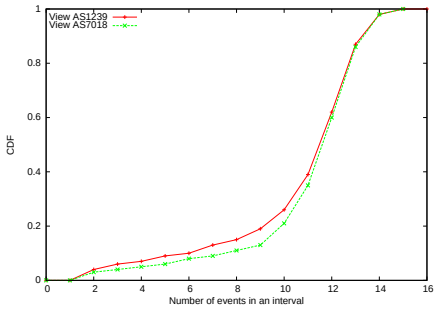


Fig. 8. Cumulative distribution of the number of events in an interval in the AUG dataset

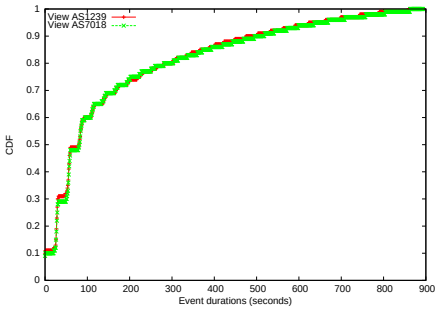


Fig. 9. CDF of event durations from the views of AS1239 and AS7018 in AUG

Next, we discuss 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. In Fig. 9, we plot the CDF of the durations for all inferred events in AUG (and both vantage points). The median duration for the inferred events is approximately 90 seconds. Note that while both vantage points might not see the same events or perhaps not similar patterns of updates for the same events, the general distribution of the duration is identical. As observed in the figure, 72% of all the events last less than 180 seconds (3 minutes). Moreover, 6% of all the events last over 600 seconds (10 minutes), which indicates that the selection of 15 minutes as our observation time interval is reasonable. 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.

We now discuss the "impact" of the inferred events. Note that "impact" can be defined either in terms of the number of ASes affected or in terms of the number of affected network prefixes. In Fig.10, we plot the distributions for both. The darker curve plots the CDF of the cluster sizes, i.e., number of ASes in each *inferred* event, while the lighter curve plots the CDF of the number of network prefixes (associated with each event). The plot uses data from the AUG dataset associated with vantage point AS 1239. The plots for vantage point AS 7018 and the SEP dataset are similar, hence we omit them here. From the figure, the median number of ASes associated with an event is 5, while the median number of network prefixes affected by any event is 11. From the figure, 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. Note that these curves do not say anything about the impact upon data traffic; it was shown in [16] that the bulk of Internet traffic is directed towards relatively few network destinations. Hence, while the impact of these events upon router load may be significant – each event generates additional updates to be processed, the impact on actual traffic is less clear.

The correlation between the size of (inferred) AS clusters and the number of network prefixes that are involved in the event is shown as a scatter plot in Fig. 11. From the figure, it is clear that the two variables are not very well correlated. This is to be expected: some ASes originate many more network prefixes than others. In addition, due to routing policies, only a subset of originated prefixes might be affected by a particular event. An interesting observation is that there are almost no events that affect only a *small* number of ASes but with a *large* number of prefixes. Hence events with small associated (origin) AS clusters (thus relatively "minor" ones among those inferred "major" events) tend to only affect reachability to a limited number of prefixes.

Finally, we present some statistics about how the "impact" of the events, using the AS cluster size, are correlated

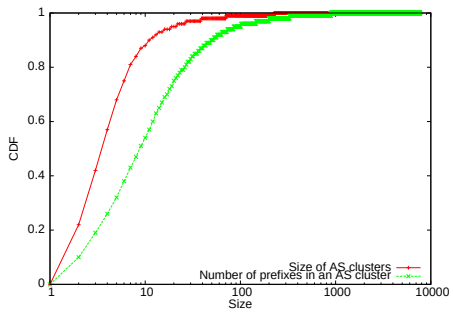


Fig. 10. CDF of "impact" of inferred events.

with its *duration*. Fig. 12 is a scatter plot; the x-axis is the duration of inferred events, while the y-axis represents the sizes of the AS clusters corresponding to the inferred events. An interesting observation here is that events that associated with a large cluster size tend to last a relatively short time (less than 300 seconds), while the longer duration events typically involve a small number of ASes. This is consistent with the observation that events with large impact tend to correspond to "simple" network events, such as link failures (or repair), session resets, etc. On the other hand, events that last longer (some of which may be "persistent") tend to involve smaller number of ASes, and possibly occur away from the Internet core. Such events might be caused by protocol oscillations [17] or unstable access links or networks in less well-managed ASes close to the edge of the Internet.

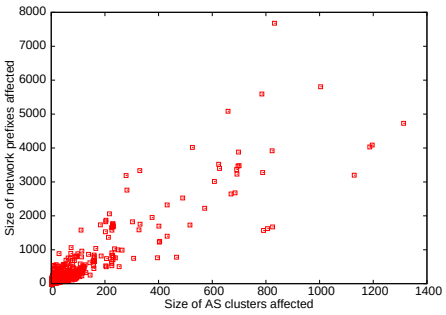


Fig. 11. Correlation between cluster size and number of affected prefixes.

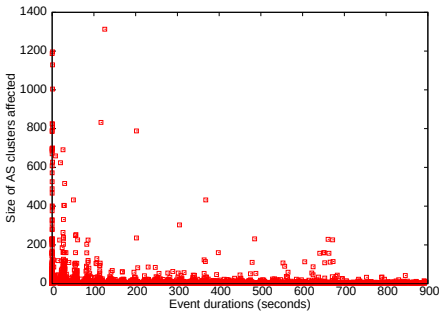


Fig. 12. Correlation between "impact" (cluster size) and event duration.

B. Corroborating Inferred 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. Note that due to the complexities of events that can affect BGP, these metrics certainly cannot cover all possible cases, but nonetheless, they provide strong evidence that our methodology is effective in identifying and separating updates associated with distinct (major) events.

1) Type of Change Analysis: When a set of ASes are affected by a particular event, it is reasonable to assume that they are *all* affected in a similar fashion. Suppose a link failure within an ISP affects a subset of its customer ASes; they are likely to be affected in the same way. For example, the paths used by the vantage point to reach these customer ASes may now be less *preferred* than those used before, e.g., they are longer because a different exit point via another AS is used. In general, it is unlikely that a single failure event causes reachability for a particular destination to *improve*, and at the same time *worsen* for a different destination [4].

Our metric is derived from the "type of change" classification discussed in [4]: for each prefix involved in the "inferred event", i.e., AS cluster, we label the set of paths observed in the duration of the event as $A_1, A_2, \ldots A_n$. Note that A_i can be an *AS Path* or a *withdrawal*, in which case, $A_i = W$. Also, we define the "stable" path A_0 as the path used to reach the destination *before* the event. Thus, the event caused the path (to the destination prefix)

to change from A_0 to A_n. Based on these observed paths, Caesar *et al.*, define five *change type* classes, which are reproduced in Table III. Based on this table, we classified *all* of the inferred events in one week ($2004/08/09$ to $2004/08/15$ in the AUG set). The distribution of events corresponding to the vantage point AS 1239 is shown in Table IV.

Type of change	Description
reroute	$A_1 \neq A_n$, $A_1 \neq W$, $A_n \neq W$
prefix-up	$A_1 = W$
prefix-down	$A_n = W$
withdraw-flap	$A_1 = A_n = W$
announce-flap	$A_1 = A_n \neq W$

TABLE III

TYPE OF CHANGE AND THEIR DESCRIPTIONS.

Reroute (%)	Prefix-up (%)	Prefix-down (%)	Withdrawn-flap (%)	Flap (%)	Total Inferred Events
556(27.8%)	314(15.7%)	185(9.3%)	23(1.2%)	920(46.0%)	1998

TABLE IV

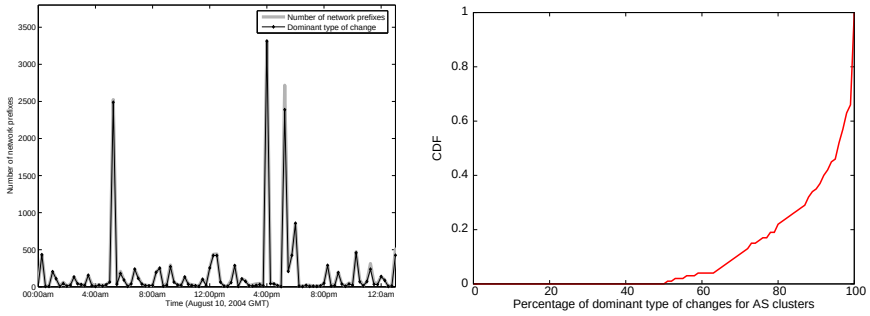
THE DISTRIBUTION OF CHANGE TYPES FOR ALL THE INFERRED EVENTS DURING THE WEEK STARTING $08/09/2004$ CORRESPONDING TO VANTAGE POINT AS 1239

If our inferred events do in fact correspond to actual network events on the Internet, it should be very likely that the set of prefixes associated with the AS clusters that we obtain be affected in the same way. In order to determine this, we study the change-type classes associated with the prefixes within a cluster. As a simple measure, we propose the *dominant change type* metric. 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.

In Fig. 13(a), we plot the size of the *dominant change type* set, along with the number of network prefixes in each *AS cluster*. In this plot, we only consider the *most significant event* in each interval. The gray curve corresponds to the number of network prefixes, while the darker curve plots the size of the *dominant change type* set. Note that the two curves are in close agreement. This indicates that, at least for the most significant event (PC_1) in an interval, the associated *AS cluster* is very likely affected by the *same* actual event. The agreement was slightly less when *all the major events in an interval were considered*, and this is illustrated in Fig. 13(b). This figure plots the CDF of the ratio of the size of the *dominant change type* set to the total number of prefixes in the cluster. In every cluster, a single *change type* accounts for at least half of the prefixes. Moreover, the dominant change type accounts for less than 80% in only 20% 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.

2) Topological Correlation: Likewise, we introduce a *topology* based metric and discuss the results of applying it on our *inferred* events. The intuition is that if a single routing event triggers BGP updates in a group of ASes at the same time, then they are likely to share some common topological properties. For example, if an access router in a backbone network fails, then we will see updates for all the customer ASes that directly (or indirectly) connect to that particular router. Building on this intuition, we introduce a topology based metric that captures the "locality" or "spatial correlation" of AS clusters within the hierarchical structure of the Internet AS topology [18].

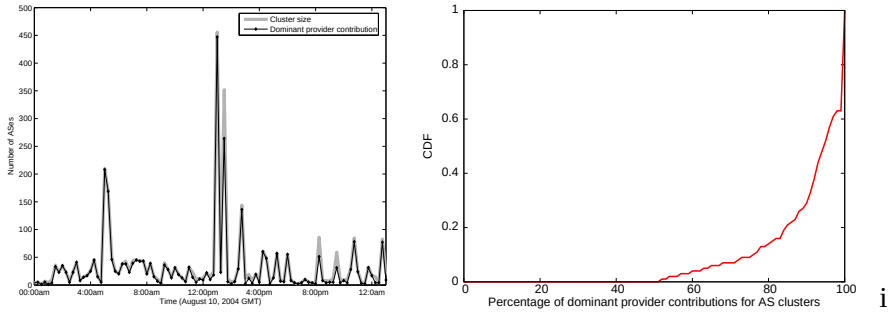
Suppose we fix the location of a vantage point and then consider its *path set*: the set of paths from the vantage



(a) Dominant change type of events extracted from PC_1 during 08/10/2204 from the view AS1239

(b) CDF of dominant change type for all inferred events in the AUG dataset from the view AS1239

Fig. 13. Change type metric



(a) Dominant provider contribution of events extracted from PC_1 during 08/10/2204 from the view AS1239

(b) CDF of dominant provider contribution for all inferred events in the AUG dataset from the view AS1239

Fig. 14. Dominant provider metric

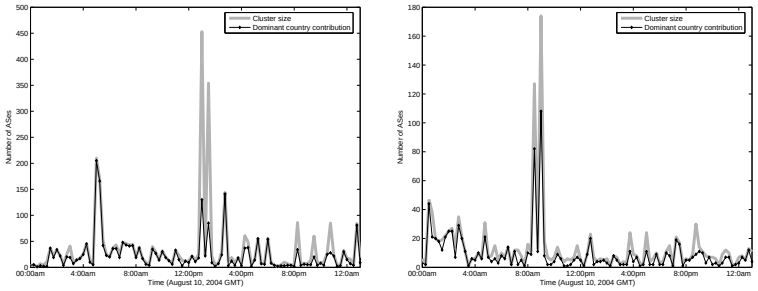
point to all other ASes on the Internet.[11] The resulting graph induced by all these paths is "tree-like", with the root being the vantage point.[12] Given this "tree", when an event affects a node (or edge), clearly the *only* ASes affected by this event are those that "lie below" the affected node. Thus, we can quantify the "contribution" of every AS to an event by enumerating the "downstream" nodes that are part of the associated *AS cluster*. Then, the *dominant provider* is the AS with the largest contribution.

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. 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*. Clearly, if a single AS[13] has contribution equal to the size of the cluster, then it is very likely that the cluster corresponds to some event that affected the particular AS.

[11]Note that we are referring to "valley free" paths, as is defined in [18]

[12]Strictly speaking, due to the effect of prefix specific policies, the structure is an acyclic subgraph; we resort to the tree analogy to simplify the description.

[13]Excluding the vantage point, which is contained in all the paths



(a) AS clusters of PC_1 (b) AS clusters of PC_2

Fig. 15. The contribution of the dominant country to AS clusters of PC_1 and PC_2 during August 10, 2004

In Fig. 14(a), we plot the *dominant provider contribution* along with the size of the AS cluster. The gray curve plots the *dominant provider contribution* while the darker line plots the size of the AS clusters. As before, this particular plot only contains the *most significant* cluster (or event) in each interval. In the figure, notice that the two curves are in agreement almost all the time, especially in the case of smaller clusters. Thus, it is reasonable to expect that in most cases, the clusters are likely to be "generated" by actual events affecting the *dominant provider* AS. In order to illustrate this metric for *all* events and not just the *most significant* ones, in Fig. 14(b), we plot the cdf of the ratio of dominant provider contribution to the size of the AS cluster. The general statistics here are similar to those corresponding to Fig. 13(b). In particular, the *dominant provider contribution* accounts for less than 80% of the size of the *AS cluster* in only 12% of the events. Thus, the AS clusters identified by our methodology are likely affected by the same network event.

3) Geographical Correlation: Intuitively if a large set of ASes are affected by the same event, then it is likely that a large fraction of the ASes lie in the same general region. Based on this intuition, we characterize the geographical locations of AS cluster through mapping the ASes within a cluster to their corresponding countries. The mapping is based on MaxMind GeoIP Country database.

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 country always has the largest contribution to the cluster. Fig. 15(a)(b) show the size of AS clusters and the contribution of the first (dominant) country associated with PC_1 and PC_2 during the observation windows on Aug. 10, 2004. As can be seen in these plots, 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.

In cases where the entire cluster maps to a single country, the results directly indicate that the different ASes might be affected by the same event. This is especially true in smaller countries where there are relatively few IXPs or PoPs. This is not quite true in larger countries (or those with dense connectivity). In such cases, a finer geographic scope, such as state or city might be useful.

Geography based metric is a limited measure of spatial correlation for two reasons. First, the mapping database can not gurantee 100% accuracy. Secondly, many ASes, e.g., backbone providers, could cover a few countries. However, a large fraction of ASes with clusters are from one or two countries. This interesting observation increases our confidence of the efficiency of PCA analysis on BGP update streams.

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. By the first

metric (*dominant change type*), we showed that the majority of prefixes are affected in the same way, and by the second and third metrics (*dominant provider and dominant country*) we established that the ASes in the cluster are strongly correlated *topologically* and *geographically* (having the same "parent" AS or in the same geographical region). 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.

VI. Case Studies

In this section, we apply our methodology to analyze six reported routing events. Of the six routing events that we have analyzed, five are reported on the NANOG mailing list since January 2003 and one is a reported maintenance outage in a backbone ISP [19]. For each event, we identify an appropriate observation interval that contains the reported time that the event occurred. We then apply our methodology on each of these intervals. A summary of the results is presented in Table V.

Known events	Time	View	# ASes observed during the window	# of inferred major events	Size of the most significant event[14]
Network outage [19]	07/21/2003	AS1221	487	11	182
Northeast blackout [20]	08/14/2003	AS11608	587	15	118
Hardware problem [21]	02/23/2004	AS6539	607	8	385
Peering link instability [22]	05/25/2004	AS11608	225	12	31
Network unreachable [23]	06/12/2004	AS1239	781	10	662
Route leaking [24]	09/17/2004	AS6539	1333	14	1168

TABLE V

Six known events used for case studies.

The first column in the table is extracted from the subject line of the e-mail that first reported the event on NANOG[15], which provides an indication of the type of events that occurred. The second column is the reported days that the events occurred. The third column denotes the particular vantage point, whose update stream is used to obtain the observation interval. The fourth column gives the number of origin ASes associated with the updates observed during that interval. Column 5 lists the number of *inferred* "major events" in the interval, and the last column shows the size of AS cluster corresponding to PC_1, i.e., the top *AS cluster*.

Clearly, for all of these reported events, there are likely other significant events that are taking place at similar time, as indicated by the number of "major events" separated by our PCA based approach and listed in the fourth column. To demonstrate that our methodology is indeed successful in separating these events, we provide a more detailed discussion on the results obtained for one of these routing events, the last event in Table. V, reported as a "route leaking" event. This event was posted by a network operator on September 17, 2004 on NANOG mailing list, stating that AS22534 is leaking its transit routes from AS3356 (Level 3) to AS6461 (Metromedia Fibre Network) [24]. In other words, AS22534 (a customer AS of both AS3356 and AS6461) is providing transit between the two providers. This in general should not happen, thus this event is probably caused by a *misconfiguration* in a BGP router in AS22534 which accidentally leaked routes learned from one provider (AS3356) to another provider (AS6461).

Based on this information, we analyze the update stream from a vantage in AS6539, which is a peer of AS6461, for the entire day. Note that AS6539 is expected to receive all prefixes advertised by any customer of AS 6461. Not surprisingly, the time series of the update volume, plotted in Fig. 16, shows dramatic increase (between 9:01 p.m. and 11:00 pm GMT) around the time that the configuration error occurred. In fact, around 10:06 pm, the vantage point (AS 6539) sees 12636 BGP update atoms associated with 1239 different *origin* ASes.

So is this dramatic increase in the update volume observed by the vantage point in AS 6539 caused *solely* by the suspected misconfiguration problem in AS22534? Or is it possible that such other events occurring concurrently also

[14]Size of the AS cluster associated with PC_1

[15]Except for the network outage event, information for which is obtained from the Sprint maintenance website.

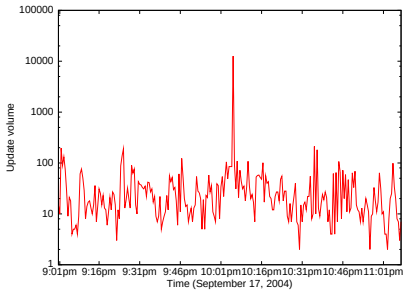


Fig. 16. BGP update streams of AS22534 leaking event from the view AS6539.

contribute to the update burst? The latter is not entirely out of the question due to the complexity of inter-domain routing. By applying our methodology to this update stream, we are able to *infer* 14 distinct events, associated with the 14 principal components which account for 80% of the cumulative variance in the update stream. Of these 14 events, only 6 events (associated with PC_1 to PC_5, and PC_9) contain origin ASes that are part of the update "spike" observed around 10:06 pm; in other words, their corresponding AS clusters intersect with the set of 1239 origin ASes associated with the updates seen during the "spike". It suggests that in addition to the misconfiguration event, there might be other events that happen to take place at the same time.

Upon closer examination, we find that the event corresponding to PC_1, which accounts for the most variation and can be considered the "most significant" event, involves 1168 different *origin* ASes. Moreover, this *AS cluster* is associated with a *reroute* type-of-change event (see Sec. V-B.1). This is consistent with what one expects to see in a "route-leaking" event: upon receiving the leak routes, AS6461 decides to switch to these routes, as in general customer routes are more preferred to other routes [25]. Hence a *predominantly large portion* of the dramatic increase in the update volume (updates associated with 1168 out of 1239 origin ASes) can be attributed to this single event. However, as we illustrate below, this single event does not explain *all the updates*.

There are 71 other *origin* ASes observed in the same burst that are not part of this *reroute* type-of-change event, and updates associated with these ASes also contribute to the increase. In particular, roughly half of these belong to the AS cluster associated with PC_2, which is of type *announce-flap*, so they are unlikely to be related to the route-leaking event. In addition, the remaining ASes correspond with the clusters associated with PC_3, PC_4, PC_5 and PC_9. Examining the AS PATH attribute in the associated updates show that they do not contain AS22534, hence they are unlikely to be part of the route-leaking event either.

The above example shows that while a single event can account for most of the updates observed at a given time, there may be several other concurrent events that contribute a smaller amount. Therefore we cannot simply attribute all updates in a BGP update "spike" to a *single* event. We have done more detailed analysis on the other case studies, which provide strong evidence for corroborating and validating our findings. Due to space limitation we will not elaborate here. In summary our case studies illustrate the effectiveness of our methodology in statistically inferring and separating BGP updates associated with distinct major events, and its utility in facilitating BGP root cause analysis and trouble-shooting by uncovering common features in the AS/update clusters and thereby narrowing down the problem space.

VII. 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 [3], [4], [6], [5]. 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 [7], 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 [26], 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.

VIII. Conclusions and Future Work

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 as well as case studies, 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. We plan to use our method to analyze real-time BGP update streams collected from various vantage points to diagnose the Internet instability.

References

[1] Yakov Rekhter and Tony Li, "A Border Gateway Protocol 4 (BGP-4)," Mar. 1995, RFC 1771.

[2] T. Griffin, "What is the sound of one route flapping?," Network 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. of ICNP*, 2003.

[4] M. Caesar, L. Subramanian, and R. Katz, "Root cause analysis of Internet routing dynamics," Tech. Rep., U.C. Berkeley Technical Report 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. of IEEE/IPIF NOMS*, Apr. 2004.

[7] R. Teixeira and J. Rexford, "A measurement framework for pin-pointing routing changes," in *Proc. of ACM SIGCOMM Network Troubleshooting Workshop*, 2004.

[8] G. Houston, "Interconnection, peering and settlements-Part I," *Internet Protocol Journal*, 1999.

[9] I.T. Jolliffe, *Principal Component Analysis (2nd edition)*, Spinger Series in Statistics. 2002.

[10] University of Oregon, "Routeviews archive project," http://archive.routeviews.org/.

[11] RIPE, "Routing information service raw data," http://data.ris.ripe.net/.

[12] M. Mao, R. Bush, T. Griffin, and M. Roughan, "BGP beacons," in *Proc. Internet Measurement Conference*, 2003.

[13] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet Routing Convergence," *IEEE/ACM Transaction on Networking*, 2001.

[14] H. F. Kaiser, "The application of electronic computers to factor analysis," *Educational and Psychological Measurement*, 1960.

[15] SSFNET, "Scalable Simulation Framework," http://www.ssfnet.org.

[16] Jennifer Rexford, Jia Wang, Zhen Xiao, and Yin Zhang, "BGP Routing Stability of Popular Destinations," 2002.

[17] Timothy Griffin and Gordon Wilfong, "On the correctness of IBGP configuration," in *Proc. ACM SIGCOMM*. 2002, ACM Press.

[18] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM*, 2002.

[19] "SprintLink scheduled maintenance and outage," `http://www.sprintlink.net/maintview/index.cgi`.

[20] NANOG, "Power outage in North East," `http://www.merit.edu/mail.archives/nanog/2003-08/msg00501.html`.

[21] NANOG, "Possible L3 issues," `http://www.merit.edu/mail.archives/nanog/2004-02/msg00794.html`.

[22] NANOG, "Level3 and Genuity," `http://www.merit.edu/mail.archives/nanog/2003-05/msg00479.html`.

[23] NANOG, "AboveNet major backbone issues," `http://www.merit.edu/mail.archives/nanog/2004-06/msg00394.html`.

[24] NANOG, "AS22534 Leaking, anybody alive their?," `http://www.merit.edu/mail.archives/nanog/2004-09/msg26114.html`.

[25] L. Gao and J. Rexford, "Stable Internet Routing Without Global Coordination," in *Proc. of ACM SIGMETRICS*, 2000.

[26] D. Andersen, N. Feamster, S. Bauer, and H. Balakrishnan, "Topology Inference from BGP Routing Dynamics," in *Proc. Internet Measurement Workshop*, Nov. 2002.