I-seismograph: Observing and Measuring Internet Earthquakes

Jun Li and Scott Brooks
Network Security Research Laboratory
University of Oregon
Email: {lijun, sbrooks}@cs.uoregon.edu

Abstract— **Disruptive events such as large-scale power outages, undersea cable cuts, or Internet worms could cause the Internet to deviate from its normal state of operation. This deviation from normalcy is what we refer to as the "impact" on the Internet, or an "Internet earthquake." As the Internet is a large, complex moving target, to date there has been little successful research on how to observe and quantify the impact on the Internet, whether it is during specific event periods or in real time.**

In this paper, we devise an Internet seismograph, or *I-seismograph*, to provide a "Richter scale" for the Internet. Since routing is the most basic function of the Internet and the Border Gateway Protocol (BGP) is the *de facto* standard inter-domain routing protocol, we focus on BGP. After defining what "impact" means with respect to BGP, we describe how I-seismograph measures the impact, exemplify its usage with several disruptive events, and further validate its accuracy and consistency. We show that we can evaluate the impact on BGP during an arbitrary period, including doing so in real time.

Index Terms—**Internet seismograph; Internet earthquake; Border Gateway Protocol (BGP); BGP impact measurement**

I. INTRODUCTION

The Internet has become a critical infrastructure of our society, yet little has been studied on how to monitor the Internet as a whole and how to quantify the impact that disruptive events may have on it. Although events such as security attacks, large-scale power outages, hurricanes, undersea cable cuts, and other kinds of natural disasters may cause observable disturbances to the normal operation of the Internet, we know little about the kind of impact each event might cause and how big it might be; the lack of such knowledge also makes it difficult to conduct effective network diagnosis, recovery, or other operation tasks. In fact, there is not even an established criteria for classifying different kinds of impacts or for quantifying what "big" or "small" means.

This paper aims to fill this gap. We have designed an Internet seismograph, or *I-seismograph*, to measure "Internet earthquakes." It not only reports the magnitude of the impact during an event period, i.e., a "Richter scale" of an Internet earthquake, but also characterizes the nature of the earthquake. During a period when everything is normal, I-seismograph will simply report zero or close-to-zero impact; during a security attack, a natural disaster, or some other large-scale incident,

if the regular operations of the Internet go awry, it can then indicate how badly the Internet got hit. Not only can we use I-seismograph to measure the impact over a period in the past, during which a disruptive event is suspected to have affected the Internet, but we also can use it to measure an Internet earthquake in real time. (Note that I-seismograph does not identify the root cause if any impact on the Internet is observed. Root cause analysis is beyond the scope of this paper.)

The main design idea of I-seismograph is hinged upon discovering the "normal" state of the Internet, and then monitoring a given period to measure how the Internet activity deviates from it. Since routing is the most basic function on the Internet and the Border Gateway Protocol (BGP) is the *de facto* standard inter-domain routing protocol, our approach uses BGP data to discover the normal and abnormal states. This presents a challenge since BGP is very dynamic and BGP data are full of outliers. Furthermore, BGP has evolved greatly over the years and the definition of normal is ever-changing. To handle this dynamic nature, we have designed a two-phase clustering method that can discover what is normal and what is abnormal over a wide time span.

In this paper, we first present our definition of impact (Sec. II). We then describe how I-seismograph addresses various challenges in order to measure the impact that BGP receives during any period (Sec. III). We not only show the results when using I-seismograph against several disruptive events (Sec. IV), but also validate I-seismograph to make sure it possesses some key properties (Sec. V). Limitations certainly exist with this work (Sec. VI), but we show I-seismograph is clearly different from the related work (Sec. VII), and our conclusions about this work are strong (Sec. VIII).

II. DEFINING IMPACT

We define an impact on BGP as any deviation from BGP's normal profile. The deviation consists of a *magnitude* and a *direction*. Assume we use a set of n distinct BGP attributes to inspect BGP, A_1, A_2, $\cdots$, A_n. Also assume we have defined a normal profile of BGP by identifying the normal values of those attributes. At any time t, if the values of these attributes of BGP are $a_1(t)$, $a_2(t)$, $\cdots$, $a_n(t)$, and they deviate from the normal profile as $\delta_1(t)$, $\delta_2(t)$, $\cdots$, $\delta_n(t)$,

the impact that BGP receives at t is then a vector as follows: $i(t) = <\delta_1(t), \delta_2(t), ..., \delta_n(t)>$.

If looking at the impact on BGP over a time window, such as during the period of an event, we can define the impact during this window, say $[t_1, t_2]$, as: $I(t_1, t_2) = \int_{t_1}^{t_2} i(t)dt$ or $\sum_{t_1}^{t_2} i(t)$, depending on whether $i(t)$ is continuous or discrete.

III. Design of I-seismograph

Having defined BGP impact as a deviation from the normal profile of BGP, we now describe how we design I-seismograph to measure it. Not only must it discover what the normal profile of BGP is, it must also be able to calculate any deviation from the normal.

A. Requirements and Challenges

I-seismograph must collect and process a very large amount of BGP data, be able to identify what data are normal and what are not, and be able to accurately quantify their difference. In doing so, it must consider both the spatial and temporal aspects of BGP. From the space dimension, BGP is a complex routing protocol concerning IP prefixes from the entire IP address space and involving BGP routers from all over the Internet. From the time dimension, the BGP protocol is constantly evolving to accommodate the growth of the Internet; accordingly, what is considered normal at one time may be abnormal at another time (and vice versa).

I-seismograph must also have good usability. Not only should it be easy to use, but it should also be flexible enough to allow for the impact calculation for any given period. It should be able to calculate the impact during a historical event, such as when the Slammer worm spread, as well as the impact that BGP is currently experiencing.

I-seismograph must also be consistent, stable, and reliable. Of key importance is that once it has sampled *enough* BGP data from different periods, the definition of the normal profile of BGP should be stable; I-seismograph should output the same impact results for a given period no matter what BGP data input it has for other periods.

We show how I-seismograph meets the first two requirements in the rest of this section, and demonstrate its consistency in Sec. V.

B. Methodology Overview

I-seismograph's basic data processing unit is BGP *databin*, which is simply a summary of the values of a set of distinct BGP attributes over a period of one minute.

To measure the impact during a monitoring period, our basic idea is to check every databin from that period, and see whether it is associated with a **normal cluster** composed of a set of normal databins, or an **abnormal cluster** composed of a set of abnormal databins. At any point there is only one normal cluster but there can be multiple abnormal clusters. The normal cluster represents the normalcy of BGP, and the abnormal clusters represent different types of BGP abnormalities. Once we know every databin's associated cluster, we then can

(a) Heavyweight mode

(b) Lightweight mode

| ○ | normal databin | m_k | monitoring period k |
| ● | abnormal databin | r_k | reference period k |
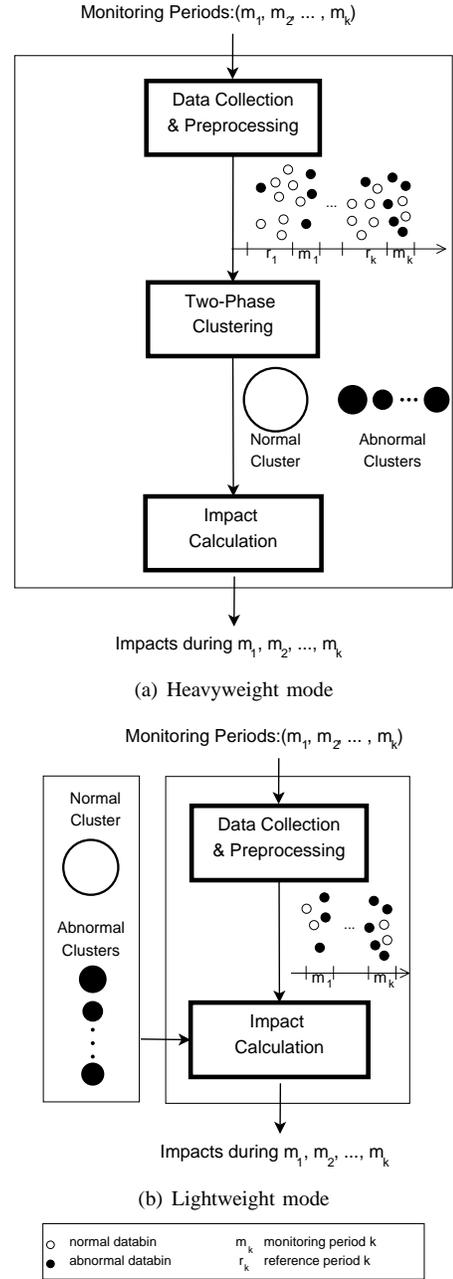
Fig. 1. Two modes of I-seismograph.

calculate the impact of the databin as well as the impact during the entire period.

I-seismograph employs two different modes for measuring BGP impact: the heavyweight mode and the lightweight mode. The two modes are depicted in Figures 1(a) and 1(b), respectively. The latter requires that the normal and abnormal clusters be known *a priori*, while the former uses an unsupervised method to discover them automatically.

Both modes include a *Data Collection and Preprocessing* component that collects BGP data and pre-processes them into distinct databins, and an *Impact Calculation* component that uses the normal cluster and abnormal clusters to calculate the impact of every databin and the aggregate impact during monitoring periods. In addition, the heavyweight mode also

includes a *Two-Phase Clustering Process* that discovers the databins which make up the normal cluster, and discovers abnormal databins and groups them into one or multiple abnormal clusters according to their similarity.

The lightweight mode is suitable for real-time Internet earthquake monitoring, or quickly checking the impact on BGP during a given period. The heavyweight mode is slower, but can be used to generate the normal and abnormal clusters that the lightweight mode will need.

C. Data Collection and Preprocessing

1) Data Collection and Cleaning: We collect BGP data from two types of periods: *monitoring periods* and *reference periods*. A monitoring period is a time window for which we want to measure the impact on BGP. It can be an arbitrary period, say $[T_1, T_2]$, that we want to monitor; or, to monitor an event that occurred from time t_1 to t_2, the monitoring period may be $[T_1, T_2]$, where $T_1 \leq t_1 \leq t_2 \leq T_2$ (as we often do not know the accurate values of t_1 and t_2, the monitoring period can be noticeably larger than the real duration of an event).

Every monitoring period is associated with a reference period. As we will see later, a reference period provides reference data to help normalize BGP data as well as run the two-phase clustering process. The reference period is chosen to be close to the monitoring period, so the BGP data from the two periods are directly comparable without worrying about data normalization. It also must be long enough to reflect the trend of BGP behavior at the close proximity to the monitoring period (we use four weeks in our experiments as we will describe in Sec. IV). While a short period may be full of outliers, a long one should only have at most a small portion, implying the majority data of the reference period can reflect what is normal during the reference and the monitoring periods. We also make sure a reference period is free of any known disruptive events to further lower its likelihood of containing too many abnormal databins. Note we do not require a reference period to only consist of normal data; instead, a reference period, like any period, can still be noisy and we need to process it.

The BGP data we collect are BGP updates. The BGP updates are the conversation records between BGP routers, and are the firsthand data about BGP. We collect BGP updates from RIPE [1] and RouteViews [2], the two organizations that maintain a number of BGP collectors. We then clean the updates by removing those that are caused by session resets between a BGP monitor and its peers. We borrow the algorithm described in [3] to filter out table dumps resulting from BGP session resets.

2) Data Organization: With the BGP updates from a given period, we convert them minute by minute into BGP databins. Because if an event has an impact on BGP, it will affect the dynamics of BGP, we choose every databin's attributes to be those that can reflect the dynamics of BGP. Based on previous studies on BGP instability and dynamics, including those from [4], [5], we have identified ten distinct BGP attributes to summarize every minute of BGP activities (Table I).

Attribute	Description
Announcement	# of BGP announcements
Withdrawal	# of BGP withdrawals
Update	# of BGP updates
WADiff	# of new-path announcements after withdrawing an old path to the same IP prefix
AADiff	# of new-path announcements to the same IP prefix (thus implicit withdrawals)
WWDup	# of duplicate withdrawals to the same IP prefix
AADupType1	# of duplicate announcements to the same IP prefix where all fields of the announcements are unchanged
AADupType2	# of duplicate announcements to the same IP prefix where only the AS-PATH and NEXT-HOP fields of the announcements are the same
WADup	# of re-announcements after withdrawing the same path
AW	# of withdrawals after announcing the same path

TABLE I
NAMES AND DESCRIPTIONS OF SELECTED BGP ATTRIBUTES.

3) Data Normalization: To discover the normal profile and different abnormal profiles of BGP, the BGP data collected for this study will span a long period (the experiments that we will report in Sec. IV involve BGP data over eight years). On one hand, we must ensure all BGP databins are comparable to each other; on the other hand, BGP is known to be evolving over time. Therefore, we must normalize the BGP databins.

Our basic idea in normalizing any given databin is to find the *baseline* value of every attribute of the databin, and then use the ratio of the original value of the attribute versus its baseline value as the normalized value of the attribute.

To find the baseline value for every attribute of a databin, our first step is to find a set of *reference databins* for the databin in question. Whether a databin to normalize is from a monitoring period or its associated reference period, we always select its reference databins from the reference period. While the majority of databins from the reference period are normal (Sec. III-C1), we must first remove outliers from the reference period. We run the K-Medoids (PAM) clustering algorithm to partition all the databins from the reference period into two clusters, and remove the databins from the cluster that is smaller—i.e., outliers. Then with the remaining databins—i.e., those belonging to the bigger cluster, we choose those databins that are of the same minute of the day as the databin in question. These databins then serve as the reference databins.

As the reference databins are from the reference period and hence their values are comparable to the databin to normalize, we simply calculate the median of each attribute of all the reference databins, and use that as the baseline value for the attribute of the databin to normalize.

D. Impact Calculation

I-seismograph calculates impact from two levels: the impact of a single databin, and the impact during a monitoring period. Its input includes a normal cluster and multiple abnormal clusters. (We describe how we obtain these clusters in Sec. III-E.) The impact of an individual databin is based on the databin's relation with the normal cluster. The impact during a monitoring period checks how all the databins from the period deviate from the normal cluster collectively.

Every databin from a monitoring period will be assigned into either the normal cluster or one of the abnormal clusters. In the lightweight mode, the procedure is straightforward: with the normal and abnormal clusters as input, I-seismograph compares every databin's distance to the medoid of every cluster—i.e., the most centrally located databin in that cluster—and assigns the databin to the cluster with the nearest medoid. In the heavyweight mode, this is achieved through the two-phase clustering which we describe in Sec. III-E.

We introduce the following concepts to measure the impact of a databin or the impact during a monitoring period:

- **Impact value (of a databin).** It measures the *distance* of a databin from the normal. We define every databin in the normal cluster has an impact value 0, and here we focus on those not in the normal cluster. Denote the databin as $d = <d_1, d_2, \cdots, d_n>$. We take the following steps: (1) For every attribute A_i ($i = 1, 2, ..., n$) of d, we use all the databins from the normal cluster to determine their mean μ_i and standard deviation σ_i of A_i. (2) We then calculate the difference between d_i and $(\mu_i \pm \sigma_i)$, denoted as δ_i. It is either $d_i - (\mu_i + \sigma_i)$ if d_i is greater than $(\mu_i + \sigma_i)$, or $(\mu_i - \sigma_i) - d_i$ if d_i is smaller than $(\mu_i - \sigma_i)$. (3) We normalize δ_i to be in the range of [0, 1] by dividing the maximum recorded value of δ_i. In the following δ_i always refers to a normalized value. (4) Then finally, we use the sum of the differences for all attributes, i.e., $\sum_{i=1}^{n} \delta_i$, as the distance of d from the normal. This distance is also called Manhattan distance. Since our study currently uses exactly 10 BGP attributes, every impact value will thus be between 0 and 10.
- **Impact direction (of a databin).** It measures the *direction* that a databin deviates from the normal. Following the discussion of impact value above and using the same notations, we define the impact direction of a databin using the deviation vector $<\delta_1, \delta_2, \cdots, \delta_n>$.
- **Impact curve (of a monitoring period).** This is the plot of the impact values of all the databins from a monitoring period over time.
- **Dominant and peak impact directions (of a monitoring period).** The abnormal cluster that has more databins from the monitoring period than any other abnormal clusters is what we call the *dominant abnormal cluster* for the period. We define the impact direction of this cluster's medoid (i.e., its most centrally located databin) as the *dominant impact direction* for the monitoring period in question. In addition, we define the impact directions of those databins from a monitoring period that have a peak impact value as the *peak impact directions* of the period. Note that those databins may or may *not* belong to the dominant abnormal cluster. The dominant direction represents the overall trend during a monitoring period, and the peak direction indicates the behavior during the maximum impact.

E. Two-Phase Clustering Process

With BGP databins from one or multiple monitoring periods, I-seismograph in heavyweight mode includes a two-phase clustering process to discover a normal cluster of normal databins and multiple abnormal clusters of abnormal databins.

The two-phase clustering is based on our concept of two-level normality: **short-term normal**, or **s-normal**; and **long-term normal**, or **l-normal**. S-normal refers to what is normal during a specific monitoring period and its associated reference period. L-normal refers to what is normal during a much longer period. Similarly, we use **s-abnormal** and **l-abnormal** to mean short-term and long-term abnormal, respectively. As such, the two-phase clustering process will take databins as input from multiple monitoring periods and their associated reference periods—which altogether spread over a long period, and process them in two different phases: **short-term clustering** and **long-term clustering**.

The short-term clustering serves as a filtering process; by discarding certain databins, it will ensure that every databin from a reference period is s-normal, whereas none of the databins from a monitoring period is. The long-term clustering then takes the result from the short-term clustering as its input, and clusters all the databins; it will discover databins that are l-normal and those that are not, and group them based on their similarity into the normal cluster and multiple abnormal clusters, respectively. Below we describe each phase in detail.

1) Short-Term Clustering Phase: We take two steps in processing the databins from a monitoring period and its associated reference period: first, we process databins from the reference period; second, we use the result to help process databins from the monitoring period.

Processing Databins from Reference Period: Assuming a reference period spans over multiple days, for each day of databins, we run a clustering algorithm, called **N-clustering**, to see if it generates an s-normal cluster that contains only s-normal databins. If it does, we retain databins from the s-normal cluster and discard all other databins; otherwise, we discard the entire day.

N-clustering is a divisive hierarchical clustering algorithm [6]. It relies on two rules: the majority rule and the tightness rule. It assumes that the s-normal cluster—if it ever exists—must consist of more than 50% of the databins from the initial input, and these databins must be tightly clustered.

As shown in Figure 2(a), N-clustering works as follows: (1) It begins with all the input databins as the root cluster, and uses K-Medoids to recursively split a cluster into two child clusters. K-Medoids is used because it creates non-overlapping clusters and is more resilient to outliers than other clustering algorithms such as K-Means. (2) Upon every split, it discards the smaller child because it has less than 50% of the databins and cannot be or lead to an s-normal cluster. (3) If the bigger child meets both the majority rule and the tightness rule, it is exactly the s-normal cluster to generate! If it meets the majority rule but not the tightness rule, it will be split again. If it does not meet the majority rule, however, no s-normal cluster will be found and N-clustering simply stops.

To determine whether or not a cluster is tight, we check its intra-distance and inter-distance [7]. The intra-distance shows how far apart databins within a cluster are, and the inter-distance is the distance between a cluster and its sibling cluster.

When the intra- and inter-distance of a cluster reaches a *knee* or inflection point, we determine that this cluster is tight and does not need to be further split. (We choose 20% as the knee since the knee typically occurs when the intra-distance becomes no more than 20% of the inter-distance.)

Processing Databins from Monitoring Period: Now that databins from the reference period are all s-normal, we further process the databins from the monitoring period to only retain those that are s-abnormal. However, doing so is more difficult than retaining s-normal databins from the reference period. In the latter, every time we split a cluster of databins into two child clusters, we can discard the smaller child since this child is guaranteed not to contain s-normal databins. Now, because the majority databins from the monitoring period could be either s-normal or s-abnormal, if we run a clustering algorithm to split databins from the monitoring period, we do not know between the bigger and the smaller child clusters, which one to discard and which to further inspect.

We overcome this difficulty by designing a new clustering algorithm, **A-clustering**, to discover a cluster of s-abnormal databins (Fig 2(b)). Like N-clustering, it is also a divisive hierarchical clustering algorithm. It begins with one initial cluster with all the databins from the monitoring period, and also uses K-Medoids to split a cluster into two new child clusters. But, every time we split a cluster, we inflate it with s-normal reference databins obtained earlier! Specifically, every time we split a cluster with n databins, including the very initial cluster, we randomly choose more than n s-normal reference databins, and inject them to the cluster, thus creating an inflated cluster. The inflated cluster will have a key property: *Its s-normal databins are the majority, and the s-abnormal databins to discover are the minority*. The majority here includes not only the injected, s-normal databins, but also those from the monitoring period that are also s-normal. As a result, after a binary split of the inflated cluster, we will be certain that the s-abnormal databins will go to the smaller child. The bigger child will not only include injected, s-normal databins, but will also act like a sticking ball to pick up as many s-normal databins as possible from the monitoring period. If the bigger child cannot pick up any s-normal databins from the monitoring period, the smaller child is already a cluster with all the s-abnormal databins and we are done; otherwise, we can continue to split the smaller child—again with s-normal reference databins injected first—until we finally find a child cluster with only s-abnormal databins.

2) Long-Term Clustering Phase: After we use short-term clustering to filter the databins for every monitoring period and its associated reference period, we can compare the databins from a monitoring period and those from its associated reference period, and see how abnormal the former are compared to the latter. However, such abnormality is based on the short-term normality, and will not indicate the impact during a monitoring period over a long term. It is also hard to compare the impact from different monitoring periods that may be far from each other.

To address this limitation, we introduce the long-term clus-

(a) N-clustering for processing databins from a *reference* period

(b) A-clustering for processing databins from a *monitoring* period
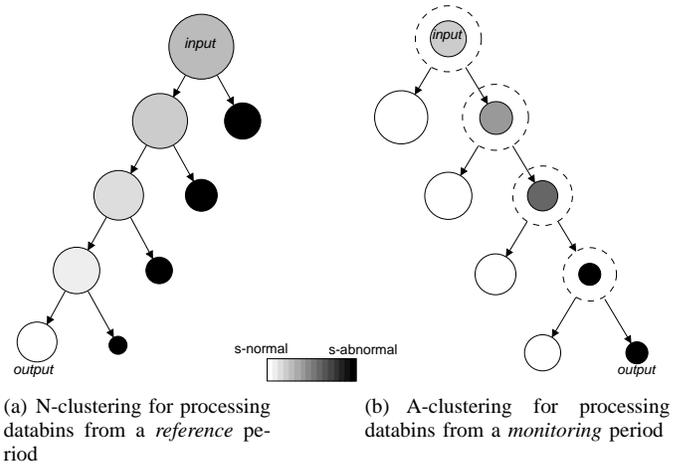
Fig. 2. Short-term clustering. (Each circle represents a cluster, the dashed circle represents an inflated cluster, and a cluster with darker shade contains a higher percentage of s-abnormal databins.)
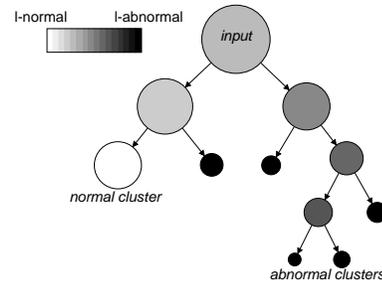
Fig. 3. Long-term clustering. (Its input is the output from the short-term clustering, and its output are the normal cluster and multiple abnormal clusters (5 in this figure)).

tering phase to derive the long-term normality, discover how abnormal the databins from different monitoring periods are from a long-term perspective, and group databins according to their long-term normality as well as long-term abnormality. The long-term clustering enables us to discover a common, long-term normality of BGP, and compare the impact from all the monitoring periods in the same context.

Same as our two short-term clustering algorithms, the long-term clustering algorithm is also a divisive hierarchical clustering algorithm. As shown in Figure 3, it will generate a normal cluster of long-term normal databins *and* multiple abnormal clusters of long-term abnormal databins. The initial input is a root cluster of all the s-abnormal and s-normal databins from multiple pairs of monitoring and reference periods. Every time we process a cluster, including the root cluster, we first check whether or not the cluster is tight by calculating its intra-distance, and compare it with the intra-distance of its parent cluster. If the two intra-distances differ by less than 1%, i.e., clustering helps little in further packing databins in this cluster, the cluster is tight, and it is a leaf cluster and we do not split it. Otherwise, we continue to use K-Medoids to split it into two child clusters. We then begin processing *every* child cluster, following the same procedure just mentioned. This recursive procedure will eventually stop, creating a tree of clusters. If the largest leaf cluster contains more than 50% of the databins

in the initial input, i.e., conforming to the majority rule, this leaf cluster is then the normal cluster; other leaf clusters are various abnormal clusters.

IV. Impact Results and Analysis

In this section, we apply I-seismograph to measure the impact on BGP, i.e., the Internet earthquake, during different monitoring periods that cover a wide time span. We report the impact results during these periods, and analyze their patterns and characteristics.

A. Setup

We have identified a number of events that may or may not have disrupted the normal operation of BGP. We selected these events from a wide time span since we want I-seismograph to apply to all times. They are Code Red worm [8], Nimda worm [9], Slammer worm [10], East Coast blackout [11], Hurricane Katrina [12], LA blackout [13], Taiwan undersea cable cut [14], Mediterranean undersea cable cut [15], and Mediterranean undersea cable cut *again* [16].

We associate every event with a two-day monitoring period that begins when the event began. I.e., we monitor an event's likely impact on BGP during a two-day period. We further associate every event with a reference period that lasts four weeks and happens exactly before the monitoring period.

B. Results Overview

We measured the impact on BGP during these nine events using the heavyweight mode, with all nine monitoring periods as the input. Our impact results include both the impact curves and the impact directions, allowing us to analyze the impact that BGP receives during each monitoring period. In the next section (Sec. V) where we evaluate the accuracy of our methodology, we further compare the results here with the impact results obtained using the lightweight mode.

C. Impact Curves

Figure 4 shows the impact curves during the nine events. We can categorize the impact curves into three categories:

- *Short-lived impacts.* The curve is typically a spike, whereas the spike may be of a high value or a low value. Except for the spike, the rest impact values are close to 0. The Code Red curve (Fig 4(b)), for example, has a small spike and the peak impact value is 0.44. The Hurricane Katrina curve (Fig 4(g)), on the other hand, has a much taller spike with the peak impact value reaching 2.75.
- *Long-lived impacts.* When the impact during a period is long-lived, the impact curve can further have different forms. It can subside from a peak value gradually over time, as shown in the Slammer curve (Fig 4(d)); or it can be bursty with numerous spikes, as in the Nimda, Taiwan and Mediterranean curves (Fig 4(c), 4(h) & 4(i), respectively). All impact curves subside toward zero as the impact diminishes, but the trend can be either gradual, as shown in the Slammer and Mediterranean curves (Fig 4(d) & 4(i)), or up-and-down, as in the Nimda and Taiwan curves (Fig 4(c) & 4(h)).

(a) Reference: Dec. 12-14, 2008

(b) Code Red Worm: Jul. 19, 2001

(c) Nimda Worm: Sep. 18, 2001

(d) Slammer Worm: Jan. 25, 2003

(e) East Coast Blackout: Aug. 14, 2003

(f) LA Blackout: Sep. 11, 2005

(g) Hurricane Katrina: Aug. 29, 2005

(h) Taiwan Cable Cut: Dec. 26, 2006

(i) Mediterranean Cable Cut: Jan. 30, 2008

(j) Mediterranean Cable Cut, Again: Dec. 19, 2008
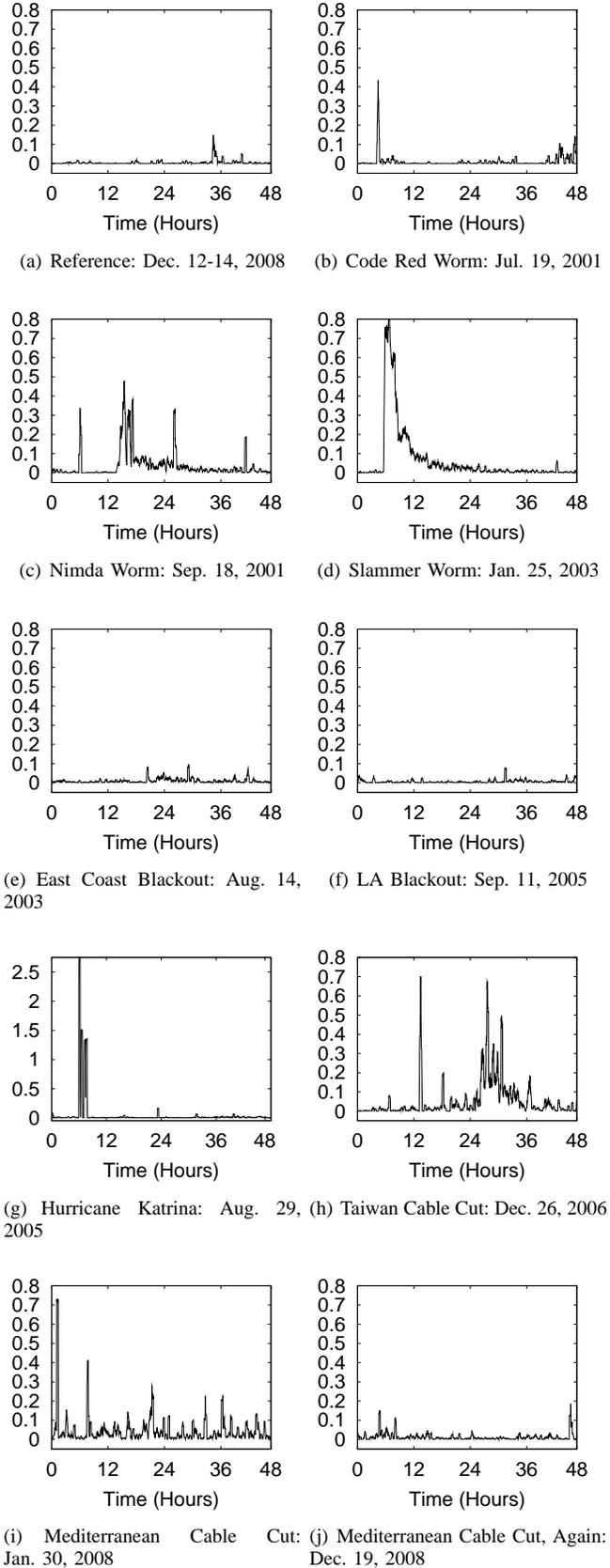
Fig. 4. Impact curves during nine different monitoring periods plus a reference period. (Every period's starting date is also shown.)

6

- *None or barely existent impact.* In this category, the impact during a period is almost none, as shown in both the East Coast and LA blackout impact curves (Fig 4(e) and 4(f)), or not much, as shown in the Mediterranean Again curve (Fig 4(j)). This category is the most frequently seen category, including most reference periods.

D. Dominant and Peak Impact Directions

An impact direction can indicate which attributes deviate from normal. Some attributes (e.g., AADiff and WADiff) show forwarding dynamics of BGP that reflect topological changes, some (e.g., WWDup and AADupType1) show pathological behavior due to redundant updates, and some (e.g., WADup) could mean both. Readers can refer to our earlier work [4] to see how we can analyze different BGP attributes to understand BGP dynamics. (See Table I for attribute definitions).

We first look at the dominant impact directions. We have found that the Slammer and Nimda worms, the Taiwan undersea cable cut and the East Coast blackout all map to the same dominant impact direction. The main BGP attributes deviating from normal include Announcement, Withdrawal, Update, AADiff, and WADiff. It shows more BGP announcements and withdrawals were sent during those periods (Announcement, Withdrawal, Update), and this higher level of dynamics is mostly due to forwarding dynamics (AADiff and WADiff). The Code Red worm, LA blackout, Mediterranean, and Mediterranean Again share another dominant impact direction, where the main deviating BGP attributes include Announcement, Update, AADiff, and AW. The BGP dynamics is mostly forwarding dynamics, but there are not extra withdrawals as in above cases. The dominant impact direction for Katrina is perhaps the most interesting, where Announce, Update, AADupType1, AADiff, WADiff, and WADup all deviate from normal, showing both forwarding dynamics (AADiff, WADiff, WADup) and pathological behavior (AADupType1, WAdup). (Note WADup could be contributing to both.)

Analyzing the peak impact directions over these periods, we found every peak corresponds to a much higher amount of BGP announcements, but a normal amount of withdrawals. Some peaks simply show a higher level of benign forwarding dynamics (the peak of Code Red, the 1st and 4th peaks of Nimda, the 1st peak of Taiwan, the 2nd peak of Mediterranean); some peaks show pathological behavior (the 2nd peak of Nimda and the 1st peak of Mediterranean); and some peaks show both (the 3rd peak of Nimda, the peaks of Slammer and Katrina, and the 2nd peak of Taiwan). These peak impact directions show the maximum impact during a period, and do not necessarily agree with the dominant impact direction.

V. VALIDATING I-SEISMOGRAPH

In this section we validate I-seismograph. In particular, we compare its heavyweight mode against its lightweight mode to see if they lead to equivalent results, and investigate its consistency to see how impact results may vary when we vary the input to I-seismograph.
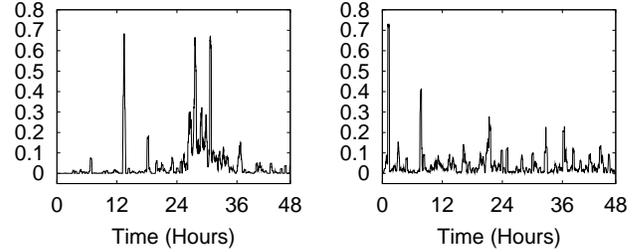
(a) Taiwan Cable Cut: Dec. 26, 2006 (lightweight) (b) Mediterranean Cable Cut: Jan. 30, 2008 (lightweight)

Fig. 5.   Impact curves produced in lightweight mode.

A. Heavyweight Mode vs. Lightweight Mode

If I-seismograph works correctly, it should generate equivalent results whether it is used in heavyweight mode or lightweight mode. In Sec. IV we have shown impact results from the heavyweight mode for nine different monitoring periods. Here, we use the Taiwan earthquake and Mediterranean cable cut as two example events, and compare their impact results from lightweight mode with those from the heavyweight mode.

We first use the other seven events as input to I-seismograph, and run I-seismograph in heavyweight mode to generate the normal cluster and abnormal clusters. Then we use the Taiwan earthquake monitoring period as the input to I-seismograph, run it in lightweight mode to obtain the impact results for the period. We also do this similarly for the monitoring period associated with the Mediterranean cable cut event.

Fig 5 shows the results for these two monitoring periods from the lightweight mode. Let us look at the difference for the Taiwan earthquake monitoring period. Between the heavyweight mode and the lightweight mode, the cumulative impact difference over the 48-hour period is 27.68, which on average is 0.010 (i.e. 27.68/(48*60)) per databin. Recall the impact value for every databin has a range from 0 to 10, this difference is clearly insignificant. The difference for the dominant impact direction is 0.116, which is the sum of the difference at every one of the ten BGP attributes. This difference is also clearly small.

The comparison of the Mediterranean undersea cable cut is more striking. Their cumulative impact difference over the 48-hour period is only 0.81, i.e., 0.0003 on average per databin. The difference between the dominant impact direction is also as small as only 0.002.

From these two events we can see that we can feed past BGP data into the heavyweight mode to generate normal and abnormal clusters, then switch to lightweight mode to more easily measure impacts for future monitoring periods, including real-time monitoring.

As the correctness of the lightweight mode is hinged upon the correctness of the normal and abnormal clusters generated from the heavyweight mode, the result above also demonstrates that the heavyweight mode is good at discovering and distinguishing the normal and the abnormal.

7

B. Consistency

Another key property that I-seismograph must possess—if its methodology is valid—is that it must be consistent with *different* input; namely, it must derive a normal cluster that defines the same normalcy of BGP as well as the same impact on BGP for any given monitoring period.

Using the same nine monitoring periods from Sec. IV, we design an iterative procedure to check the consistency:

1) Pick n random permutations of the nine monitoring periods, and repeat steps 2) and 3) below for every permutation. There are 362,880 different permutations, and we randomly pick $n = 40$ from them.

2) Denote the current permutation $m_{x1}, m_{x2}, \cdots, m_{x9}$. Run I-seismograph in heavyweight mode nine times: first time with m_{x1} as the only monitoring period, then every following time add the next monitoring period in sequence, until the last time that includes all nine monitoring periods.

3) Each time after adding a monitoring period, compare the results from I-seismograph in terms of three consistency metrics (see below), and record the difference.

4) Gather all the stepwise consistency check results from 3) and conduct the statistical analysis to see if I-seismograph has the consistency property.

We use the following three consistency metrics:

- *Normal cluster difference.* Assuming the derived normal clusters from two different runs of I-seismograph are N and N', their difference is the sum of their difference along each BGP attribute A_i ($i = 1, 2, ..., n$). Assuming the mean and the standard deviation of A_i for N's databins are μ_i and σ_i, and those for N' are μ_i' and σ_i', the difference of N and N' along A_i is $\frac{1}{2}|(\mu_i' + \sigma_i') - (\mu_i + \sigma_i)| + \frac{1}{2}|(\mu_i' - \sigma_i') - (\mu_i - \sigma_i)|$.
- *Impact curve difference.* Assuming the impact curves for a monitoring period $[t_1, t_2]$ are $i(t)$ and $i'(t)$ from two different runs of I-seismograph, their difference is $\sum_{t_1}^{t_2} |i'(t) - i(t)|$.
- *Dominant impact direction difference.* Assuming the dominant impact directions for a monitoring period $[t_1, t_2]$ are d and d' from two different runs of I-seismograph, their difference is the sum of d and d''s absolute difference along each attribute.

Fig 6 shows our results. Clearly, as more monitoring periods are added, all three consistency metrics converge to 0, meaning our results will be consistent or almost the same when even only a small number of monitoring periods are used.

VI. DISCUSSIONS AND OPEN ISSUES

Performance of I-seismograph. The performance of I-seismograph depends entirely on the mode of operation. Even on a laptop computer (Intel Core 2 Duo T5550 @ 1.83 GHz, 3GB RAM), it takes less than a minute for I-seismograph in lightweight mode to analyze the impact of a two-day period. I-seismograph in heavyweight mode would need much longer time, approximately 40 minutes for each two-day period when processing a total of nine such periods simultaneously. However, I-seismograph in heavyweight mode can run offline, and does not need to be invoked often.
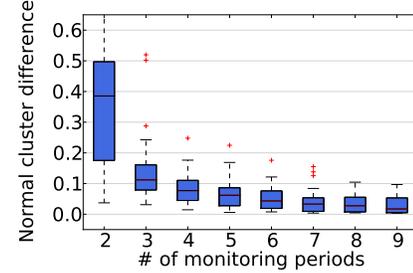
(a) Normal cluster difference
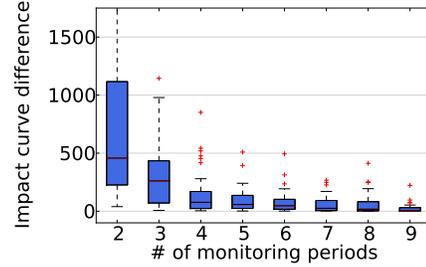
(b) Impact curve difference
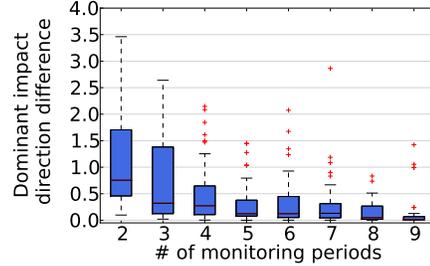
(c) Dominant impact direction difference

Fig. 6. The consistency of I-seismograph.

Data Sources. I-seismograph currently relies on BGP collectors from RouteViews and RIPE to gather BGP updates as its input. As these collectors are in specific locations and probably cannot collect all the BGP updates over the Internet, they probably cannot provide I-seismograph with the most comprehensive view w.r.t. how BGP may be affected. One investigation is to study how the observed impact changes when the input is from different sets of BGP collectors.

What does it mean when BGP receives an impact? Depending on which BGP attributes deviate from the normal state, receiving an impact is not necessarily a bad thing! For example, while a lot of WWDup is pathological, a higher number of BGP updates could simply mean BGP is doing its job.

Receiving an impact during an event does not necessarily mean that the impact is caused by the event, either. There could be other things happening simultaneously that cause the impact. While I-seismograph can report the impact that BGP is experiencing, it cannot replace root cause analysis on what exactly has caused the impact.

I-seismograph and root cause analysis. On the other hand, I-seismograph probably can facilitate root cause analysis by

providing key information such as the attributes of BGP that look abnormal, the pattern of the impact curve during a period, and even the specific databins that map to an abnormal impact. It may even further help trace which autonomous systems or prefixes contributed or were involved when BGP experiences an unusual impact. Designing an interface between I-seismograph and all such usage is therefore very useful.

Deploying I-seismograph. We have set up I-seismograph for real-time monitoring of Internet earthquakes, and we are making it available through our web site (http://netsec.cs.uoregon.edu/research/rf.shtml). It will be interesting to see how it performs in real situations. We will be able to learn, for example, how often we need at least to update the normal and abnormal clusters. As we can update normal and abnormal clusters offline and can obtain BGP data in real time through services such as BGPMon [17], we can use I-seismograph as a smooth online monitoring tool.

VII. RELATED WORK

Monitoring the routing infrastructure, especially BGP, has largely focused on its dynamics such as instability or pathological behavior. Researchers have not only measured BGP dynamics (e.g., [18], [4]), but have also attempted to investigate their origin (e.g., [19], [20]). There are also tools such as BGPlay [21], iBGPlay [22], and LinkRank [23] to visualize BGP dynamics. However, none of these studies or tools can help quantify how much the routing infrastructure, or BGP in particular, deviates from its normal state when certain dynamics happen. In fact, because most previous BGP measurement work focuses on a specific period, they do not even offer what normal might be in a long-term sense.

Many works (e.g., [24], [11], [14]) also investigated the effects of certain Internet worms, electricity outage, or undersea cable cut and other events on BGP. These works discovered that the Internet could experience a much higher level of dynamics under severe conditions. As every investigation is specific to a specific event, these studies cannot be unified to provide a uniform approach to measuring the impact on BGP.

In our own previous studies, we have also investigated an Internet routing forensics framework to try to detect and classify anomalies of BGP when certain events occur. Research in [25] also analyzes BGP updates to detect several types of anomalies. All these studies did not attempt to *quantify* those anomalies and make them comparable across different periods.

VIII. CONCLUSIONS

While the Internet is a critical infrastructure of our society, little has been done to monitor it as a whole and report the impact—or what we call an "Internet earthquake"—that it may be experiencing at any time. The fact that the Internet is a large, complex moving target makes this task challenging.

To address this problem, we devised a measurement tool called I-seismograph. It focuses on the most essential function of the Internet—routing, and the *de facto* inter-domain routing protocol—BGP. It uses a two-phase clustering method to discover the normal and abnormal states of the Internet, measures how much the BGP dynamics deviate from its normalcy during any time, and reports both the magnitude of the deviation—i.e., the "Richter scale" of an Internet earthquake—and the direction of the deviation.

I-seismograph is easy to use, and can measure an Internet earthquake either during an arbitrary period from the past or in real time. We have demonstrated its usage and shown the results from applying I-seismograph during different monitoring periods. We have also validated it, and found it is both accurate and consistent.

REFERENCES

[1] RIPE NCC, "RIPE routing information service raw data," http://data.ris.ripe.net/.
[2] Univ. of Oregon, "Route Views Project," http://www.routeviews.org/.
[3] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang, "Observation and analysis of BGP behavior under stress," in *Proceedings of ACM IMW*, November 2002.
[4] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz, "BGP routing dynamics revisited," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 2, pp. 7–16, April 2007.
[5] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Trans. on Networking*, vol. 6, no. 5, pp. 515–528, 1998.
[6] J. Han and M. Kamber, *Data Mining: Concepts and Techniques, 2/e.* Morgan Kaufmann Publishers, 2006.
[7] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *ACM SIGCOMM '05*, 2005, pp. 217–228.
[8] Computer Emergency Response Team, "CERT advisory CA-2001-19 Code Red worm exploiting buffer overflow in IIS indexing service DLL," http://www.cert.org/advisories/CA-2001-19.html, July 2001.
[9] ——, "CERT advisory CA-2001-26 Nimda worm," http://www.cert.org/advisories/CA-2001-26.html, September 2001.
[10] ——, "CERT advisory CA-2003-04 MS-SQL server worm," http://www.cert.org/advisories/CA-2003-04.html, January 2003.
[11] J. Cowie, A. Ogielski, B. Premore, E. Smith, and T. Underwood, "Impact of the 2003 blackouts on Internet communications," http://www.renesys.com/news/2003-11-21/Renesys_BlackoutReport.pdf, November 2003.
[12] "Hurricane Katrina chronology of events," http://www.fpl.com/storm/katrina_chronology.shtml.
[13] R. C. Archibold, "Accident causes blackout in much of Los Angeles," in *The New York Times*, September 12, 2005.
[14] S. LaPerrire, "Taiwan earthquake fiber cuts: a service provider view," in *NANOG 39*, February 2007.
[15] E. Zmijewski, "Mediterranean cable break," in *Renesys blog*, January & February 2008.
[16] I. Publishing Ltd, "Three undersea cables cut: traffic disturbed between Europe and Asia," December 19, 2008.
[17] H. Yan, R. Oliveira, K. Burnett, D. Matthews, L. Zhang, and D. Massey, "BGPmon: A real-time, scalable, extensible monitoring system," in *Proceedings of Cybersecurity Applications and Technologies Conference for Homeland Security*. IEEE Computer Society, 2009.
[18] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," in *ACM SIGCOMM*, 1997, pp. 115–126.
[19] D. Chang, R. Govindan, and J. Heidemann, "The temporal and topological characteristics of BGP path changes," in *Proceedings of ICNP*, November 2003, pp. 190–199.
[20] A. Feldmann, O. Maennel, Z. Mao, A. Berger, and B. Maggs, "Locating Internet routing instabilities," in *ACM SIGCOMM*, August 2004.
[21] L. Colitti, G. Battista, I. Marinis, F. Mariani, M. Pizzonia, and M. Patrignani, "BGPlay," http://www.ris.ripe.net/bgplay.
[22] "iBGPlay," http://www.ibgplay.org.
[23] M. Lad, L. Zhang, and D. Massey, "Link-rank: A graphical tool for capturing bgp routing dynamics," in *IEEE/IFIP NOMS*, 2004.
[24] J. Cowie, A. Ogielski, B. Premore, and Y. Yuan, "Internet worms and global routing instabilities," in *Proc. of SPIE International symposium on Convergence of IT and Communication*, July 2002.
[25] B. A. Prakash, N. Valler, D. Andersen, M. Faloutsos, and C. Faloutsos, "BGP-lens: Patterns and anomalies in Internet routing updates," in *Proc. of ACM SIGKDD*, 2009.