Wrinkles in Time: Detecting Internet-wide Events via NTP

Meenakshi Syamkumar*, Sathiya Kumaran Mani*, Ramakrishnan Durairajan†, Paul Barford*‡ and Joel Sommers§
*University of Wisconsin-Madison, †University of Oregon, ‡comScore, Inc, §Colgate University

***Abstract*—Understanding the nature and characteristics of Internet events such as route changes and outages can serve as the starting point for improvements in network configurations, management and monitoring practices. However, the scale, diversity, and dynamics of network infrastructure makes event detection and analysis challenging. In this paper, we describe a new approach to Internet event measurement, identification and analysis that provides a broad and detailed perspective without the need for new or dedicated infrastructure or additional network traffic. Our approach is based on analyzing data that is readily available from Network Time Protocol (NTP) servers. NTP is one of the few on-by-default services on clients, thus NTP servers have a broad perspective on Internet behavior. We develop a tool for analyzing NTP traces called *Tezzeract*, which applies Robust Principal Components Analysis to detect Internet events. We demonstrate Tezzeract's efficacy by conducting controlled experiments and by applying it to data collected over a period of 3 months from 19 NTP servers. We also compare and contrast Tezzeract's perspective with reported outages and events identified through active probing. We find that while there is commonality across methods, NTP-based monitoring provides a unique perspective that complements prior methods.**

I. Introduction

Unexpected events in the Internet can impact users in a variety of ways. On one end of the spectrum are small-scale events such as localized route changes that occur for any number of reasons and that cause only a brief increase in latency for users. At the other end of the spectrum are large-scale events such as outages (*e.g.,* the Baltimore Howard Street Tunnel fire [1]) that can cause wide-spread service disruptions and impact many users for many days.

Understanding the characteristics of unexpected events such as their prevalence and impact is important for planning, configuration and management of networks. It can illuminate weaknesses and vulnerabilities in network design and implementation. It can also clarify how measurement, monitoring and diagnostic capabilities might be deployed more effectively and efficiently. Furthermore, real-time event detection offers the opportunity to identify the scope and details of events and restore service in a timely fashion.

There are a number of challenges in developing the capability to identify and understand network events. First is the problem of gathering measurements that can provide a sufficient reach and detail in an efficient and timely fashion. Second is the problem of detecting and localizing the scope of events within a potentially vast amount of measurement data in an accurate and reliable fashion. Third is the problem of event diagnosis that can lead to effective and efficient remediation. Prior methods for Internet event detection (*e.g.,* [2]–[5]) typically rely on data from a dedicated measurement infrastructure, and can suffer from noise that is inherent in data collected in the Internet.

In this paper, we address the problem of *Internet event detection*. We define an "Internet event" as a sudden change in conditions that manifests as a change in packet latency experienced by a cluster of clients. The goal of our work is to develop a method for Internet event detection that can provide both a broad and detailed perspective without the need for deployment of new or dedicated measurement infrastructure. Several prior studies have utilized *existing* wide-area infrastructure for path failure monitoring and latency change detection (*e.g.,* [6], [7]). We adopt a similar approach, however real-time monitoring is beyond the scope of our current work.

Our method for Internet event detection utilizes a new source of data: measurements from Network Time Protocol (NTP) servers. NTP is one of the few on-by-default services on clients and it is ubiquitously deployed, thus NTP data can provide a broad perspective on Internet state. The efficacy of extracting latency measurements from NTP data and the diverse coverage of clients provided by NTP servers was demonstrated in [8]. Our study utilizes an expanded technique to extract one way delays (OWDs) between NTP clients and servers [9].

We develop a technique for detecting Internet events from OWDs extracted from NTP data. Intuitively, a jump in OWDs between a cluster of clients and a server is an indication of an event. Our technique is based on applying Robust Principal Component Analysis (RPCA) [10] to OWDs for client aggregates (defined by network prefixes) *ex post facto*. Events are identified when specified thresholds are exceeded as explained in §III. This approach enables the scope, duration and other details of events to be identified. RPCA is attractive for our application since it is more resilient to noisy data than standard PCA [11].

Our RPCA-based event detection technique is realized in a tool we call *Tezzeract*. We conduct a sensitivity analysis to establish a configuration for Tezzeract that will provide consistent and reliable results. Next, we demonstrate Tezzeract's ability to detect events through a set of controlled experiments, by injecting randomized events into NTP traces. Following that, we report on the results of events identified in ~1B NTP transactions collected over a period of 3 months from 19 NTP servers in the US. We find that the average number of events detected per day varies by NTP server and the size of its client population. We also find that the number of events detected per day varies based on whether client-to-server (c2s) or server-to-

ISBN 978-3-903176-08-9 © 2018 IFIP

client (s2c) OWDs are considered. Finally, we examine events that are detected by multiple servers, and find that as many as 10 out of the 19 traces may show an event simultaneously.

We compare and contrast the events identified by Tezzeract with two other data sources including *(i)* the ongoing Internet-wide Census and Survey project at ISI [12] and *(ii)* public reports of actual outage events. The comparisons are not intended to "validate" our method since no ground truth for Internet-wide events is available (an exception being reported events). Rather they are meant to demonstrate the utility of our method and how it compares and contrasts with other detection methods. In the case of the comparison with ISI's active probe data, we find that Tezzeract is more conservative, reporting many fewer events per day. This can be attributed to the liberal definition of event used in [12], which is simply missing ping measurements. However, Tezzeract does detect up to 67% of the events identified in the ISI data. Finally, comparison with a reported outage shows that Tezzeract effectively identifies the event and the underlying OWDs used to drive the analysis provide a useful perspective on the event's impact.

In summary, this paper makes the following contributions. *(1)* We introduce the use of NTP traces for Internet event detection, which enable broad and detailed analysis without the need for dedicated measurement infrastructure; *(2)* We describe a new method for Internet event detection based on applying RPCA to latency measurements from client clusters, which is implemented in a tool called Tezzeract; *(3)* We demonstrate the efficacy of NTP traces and our method by reporting on controlled laboratory experiments, applying Tezzeract to a large NTP data corpus, and comparing and contrasting with events detected by other methods.

We find that while there is commonality across methods, NTP-based monitoring provides a perspective that is unique, accurate and complements prior methods.

II. Datasets

A. NTP data

The NTP is both a protocol and a global hierarchy of reference servers. At the top of the server hierarchy, referred to as stratum 0, are high-precision time sources such as GPS-based and atomic clocks. These servers act as highly accurate references for servers in the next level of the hierarchy, stratum 1, which are also known as *primary* servers. *Secondary*, stratum 2 servers synchronize from stratum 1 servers and so forth down to stratum 15, which is the lowest level of the hierarchy. For redundancy, servers may also peer with others at the same level.

NTP clients compute a precise time estimate by synchronizing with one or more than one servers. Hosts running a commodity operating system are typically configured to synchronize with a default NTP server(s) (*e.g.*, `time.windows.com`, `time.apple.com`, `0.pool.ntp.org`), but can be configured to use a specific NTP server or set of servers. NTP hosts or clients typically connect to reference clocks that are stratum 2 or higher. Lists of stratum 1 and stratum 2 servers are maintained by `ntp.org`. Synchronization from these servers typically requires permission from the server administrators.

We assembled the dataset used in our study from NTP servers that are listed as part of `pool.ntp.org`. We started by reaching out to several NTP operators and explained our research goals; several operators responded positively. Out of the many who responded, we carefully selected eight NTP operators who maintain 19 different servers and obtained datasets in the form of full packet (libpcap) traces.

An intrinsic component of NTP (and in turn in the traffic captured at the servers) is the presence of timestamps in packets that are exchanged between NTP clients and servers. In particular, four timestamps are included in the NTP packets that are exchanged as part of the NTP synchronization procedure (known as *polling*): $t0$, the time at which a clock synchronization request is sent; $t1$, the time at which the request is received at the NTP server; $t2$, the time at which the response is sent by the server; and finally $t3$, the time at which the response is received by the client. We use these four timestamps to calculate the client-to-server (c2s) and server-to-client (s2c) one-way delays (OWDs). The NTP protocol running on clients determine the *polling interval* (in seconds), which is the period between NTP packets sent to a server.

Unfortunately, the captured packets have no explicit information about the level of synchronization of client(s) with NTP server(s). As a result, we must identify and remove packet exchanges between clients and servers in which the clients are observed not to be in synchronization (otherwise OWD estimates would be inaccurate). We utilize a filtering method described in [9] that employs NTP-specific heuristics on extracted OWD values, polling intervals and NTP packet fields, and divides clients into various *precision tiers* based on inferred synchronization quality. We use the OWDs from only the highest precision tier *i.e.,* clients that exhibit tight synchronization with NTP servers.

TABLE I
Summary of NTP traces used in this study.

Server ID	Server Organization	Total Measurements	Total Unique Clients	Client Prefixes [Fraction]
AG1	Independent	36,309,416	171,326	19,633 [7.6e-04]
CI1	ISP	1,483,460	549	158 [5.4e-06]
CI2	ISP	780,580	342	145 [3.5e-06]
CI3	ISP	1,305,499	357	173 [3.2e-06]
CI4	ISP	665,732	240	96 [3.5e-06]
EN1	ISP	727,873	260	140 [3.8e-06]
EN2	ISP	813,531	229	106 [3.5e-06]
JW1	Commercial	2,394,120	3,318	1,377 [4.5e-05]
JW2	Commercial	2,914,157	3,874	1,567 [3.8e-05]
MW1	University	1,441,746	10,232	33 [2.8e-05]
MW2	University	40,129,376	49,179	18,369 [1.1e-03]
MW3	University	8,514,328	2,844	463 [2.2e-05]
MW4	University	24,864,872	45,717	17,547 [6.3e-05]
MI1	Commercial	847,884,900	641,378	42,820 [5.1e-04]
PP1	Independent	800,791	6,928	2,321 [1.9e-04]
SU1	ISP	65,733,781	1,029,575	57,942 [1.3e-03]
UI1	University	26,921,525	18,951	519 [1.22e-05]
UI2	University	51,722,823	22,462	1012 [2.12e-05]
UI3	University	46,321,161	22,351	674 [2.57e-05]

Table I summarizes the key characteristics of the NTP data which forms the basis of our study. The NTP servers are located in 9 different cities, and include a combination of *(1)* 2 different Internet service providers in Chicago (IL), Edison (NJ), and Salt Lake City (UT) resulting in 7 NTP servers, *(2)* 3 commercial NTP servers in Jackson (WI) and Monticello (IA), *(3)* 7 university campus NTP servers in Madison (WI) and Urbana-Champaign (IL), and *(4)* 2 independent/community NTP servers in Atlanta (GA) and Philadelphia (PA). In the table, we observe a wide variation in the number of measurements gathered from each server, as well as a wide range of number of unique clients. Note that all measurements and clients in our study are from IPv4-based networks. In the table, we also include the number of IPv4 prefixes that contain the client population, as well as the fraction of total routable prefixes. The prefix data used to compute the rightmost column in the table comes from CAIDA [13].

B. Address prefix data

Our technique for Internet event detection depends on grouping clients into IP address prefix clusters. The address prefix data that we use is collected as part of CAIDA's prefix-to-AS (Autonomous System) mapping from the RouteViews project [13]. We also use the the IP-to-AS mapping data from Team Cymru [14] to enrich our perspective on widely used prefixes.

C. Datasets for comparative analysis

We use datasets from a number of other efforts to provide perspective on network events that we identify using Tezzeract. As noted above, *validation* is challenging due to the lack of reliable ground truth information. Thus, we draw on two sources in an attempt to understand and contextualize the events detected through our framework. Our goal is to use these comparisons in a targeted fashion to highlight how NTP-based event detection can provide an important and useful and complementary perspective on network events. Specifically, we use *(i)* Internet outage data from ISI's census and survey project [12], [15], and *(ii)* events reported on the outages mailing list [16]. These datasets were all collected contemporaneously with our NTP data and offer a broad perspective about the events that we identify in the NTP logs. Specifically, ISI's census and survey offer a network operation and configuration perspective, whereas events reported on the outages mailing list offer a (limited) operator perspective.

III. Methodology

To identity the events in NTP logs, we developed a framework and implemenation called *Tezzeract* which has two main objectives. The first objective is to identify *all events* in NTP logs, where an *event* is defined as a *significant* change in OWD that affects multiple clients within an IP prefix. Most events such as outages and route changes will manifest in a large increase in OWDs, while other events such as peering updates could manifest in a decrease in OWDs. The second objective is to provide details on characteristics of events in terms of duration, number of clients that experience an event on a per server basis and across servers. To achieve these objectives, Tezzeract consists of two algorithms: *TezzeractClusterGenerator* and *TezzeractEventDetector*, which we describe below.

A. Cluster generator

Tezzeract begins by ingesting OWD data from tightly synchronized clients[1] to generate clusters of NTP clients in a matrix. A *cluster* is simply the largest IP prefix aggregate in which we observe a given NTP client. The *TezzeractClusterGenerator* algorithm takes three inputs: *(a)* NTP logs from tightly synchronized clients, *(b)* IP prefix-to -AS mapping from CAIDA [13], and *(c)* IP address-to-AS mapping from Team Cymru [14].

The algorithm starts by extracting the IP addresses of clients from NTP logs and creates prefix tries using IP prefix/address-to-AS mapping datasets. Next, for every client C that synchronizes time with an NTP server S, the longest matching prefix among the CAIDA and Team Cymru data sources is determined. If a prefix is not already seen, a new cluster is created with the prefix as key and the set of clients for the new cluster is initialized with C. Otherwise, C is added to the set of clients of an existing prefix cluster. The clustering process accomplishes two goals: *(1)* it creates client groupings which naturally relate to Internet routing and management activity, which we hypothesize are commonly related to observed outages and performance disruptions, and *(2)* it reduces the number of dimensions of the matrix on which event detection is applied (see §III-B), thus reducing computational demands.

B. Event identifier

The *TezzeractEventDetector* algorithm generates a matrix of OWD values for every observed prefix cluster. Specifically, for every prefix cluster P, an OWD matrix of dimension t x n is generated, where t is the time bin used to group every client in a row and n is the number of clients in a prefix cluster. From the NTP logs, the polling intervals are extracted and t is determined using the median of minimum polling intervals, which determines the frequency of NTP packet exchanges for individual clients. Subsequently, the start and end epochs are generated from timestamps. Using the epochs, the time dimension of the matrix is determined and an empty matrix is generated.

Next, the algorithm populates the t x n matrix using the OWDs extracted from the NTP packets. Note that our description focuses on a single matrix for brevity. There are actually *two* matrices constructed (and processed in later steps): one for client-to-server (c2s) OWDs and one for server-to-client (s2c) OWDs. For each client C, the corresponding OWD vectors and epoch timestamps are extracted. These timestamps are used to determine the value stored at a particular index in the matrix for a client. If a client has multiple OWDs in a particular time bin, we take the maximum value. In our algorithm, if a particular client has no value for a given time bin due to missing NTP packets, we leave that entry as "NA". We remove all those prefixes with n less than 2 as well as all rows with complete NA values, resulting in a t' x n matrix, where $t' \leq t$.

[1]NTP clients that have tight synchronization with their servers by accurately accounting for and correcting the clock drift. We utilize the technique described in [9] to identify such clients.

The core of our event detection method is based on identifying outliers in the t' x n matrix. We do this by applying Robust Principal Component Analysis (RPCA) [17] with Mahalanobis distance-based thresholding [18]. RPCA is robust to missing data, which is a key aspect of our matrix [19]. This approach allows us to identify events and to characterize them in terms of time duration and number of clients on every individual OWD matrix. We begin by establishing centers of the OWD vectors from the matrix that are projected on an ellipse. Subsequently, the principal components are derived as the eigenvectors of the robust covariance matrix of the projected data points.

RPCA Calibration. Informed by the prior work on the sensitivity of PCA for detecting network anomalies [20], we use cautions in selecting configuration parameters. In particular, the following three parameters are important:

- **Number of principal components** (top_k) that are used for the new projection to reduce dimensions. If we use all n principal components derived from a t' x n matrix, then we are able to account for maximal variance in the data. In this study, we consider the algorithm's sensitivity to choosing the top_k principal components.
- **Scoring distance threshold**, which is also known as the Mahalanobis distance. Similar to prior efforts in this space, we set the outlier detection threshold based on ROBPCA [18], which provides the statistical reasoning for choosing threshold values.
- **Orthogonal distance threshold**, a distance metric that is needed *only* if we choose fewer than n principal components. For k principal components ($k < n$), the value of the threshold is established using ROBPCA [18].

Given n principal components, each of which contributes to some percentage of variance in the NTP data, we select the top_k components with *non-negligible* variance. In our analysis, we considered different variance threshold values for determining how to select the top_k principal components. Specifically, we first select NTP logs from two random days and compute the variance contributed by every principal component for all the clusters across all the servers. Next, we iteratively determine thresholds by ignoring principal components that produce lower variance than the current threshold. The threshold values iterated are 0.5, 1, 2, 5, 10 and 20 (and all thresholds are percentages).

Figure 1 shows the variance threshold along with changes in the number of events detected (bottom) and prefix clusters affected (top) for the MW3 NTP server.[2] From this plot for the MW3 server, we consider a threshold between 5 and 10% to represent a reasonable tradeoff between being too sensitive and treating too many OWD fluctuations as significant events on the one hand, and ignoring what are likely to be important performance disruptions on the other hand. Based on this analysis, we set the top_k variance threshold to be 5% in all the

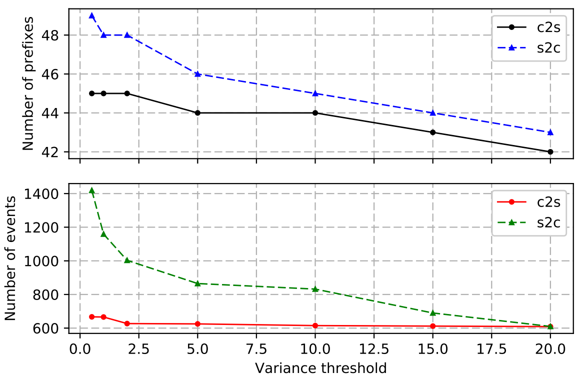

Fig. 1. **Sensitivity analysis of top_k parameter for MW3 server.**

analyses presented in this paper. Finally, the score estimates for determining events are calculated using Mahalanobis distance and Orthogonal distance in the projected space. The outliers are identified by applying thresholds on the scores using the thresholding technique similar to ROBPCA [18].

RPCA Implementation. To implement the RPCA component in the event detector, we use the PcaNA R package [21] which consists of many RPCA implementations including Projection Pursuit (PP) [22], Elliptical PCA (EPCA) [10], PCAGRID [23], ROBPCA [18] and Robust Covariance Estimator (RCE) [17]; both ROBPCA and RCE use Minimum Covariance Determinant (MCD). In particular, we use the RCE with MCD as our RPCA for two reasons. First, prior efforts have shown the threshold effectiveness of MCD in ROBPCA on datasets with missing values [24], which is also an issue with our data. Second, in a limited comparison experiment, we evaluated each of the RPCA implementations using a subset of data with validated events and compared the scores. We found that the scores from other methods were either inconsistent or failed to identify the events of interest in comparison with RCE. Furthermore, for the same data, we found that the runtime performance of RCE (0.74s) is over $\sim$3x faster than the other RPCA methods: EPCA (2.26s), PP (2.83s) and PCAGrid (3.09s). The result is an efficient implementation of *Tezzeract*, where run time on the largest NTP log for one day was 683.79 seconds.

IV. Results

A. Synthetic event tests

To illustrate the efficacy of Tezzeract's event detection capability in a controlled and repeatable fashion, we conduct a series of tests by injecting synthetic events into NTP traces. Our test data is based on an NTP log from a single server (AG1) for a single day (November 15, 2015). We run Tezzeract on the base trace and mark all events detected so they can be removed from further test results. We then modify entries in the trace to simulate events. We consider two factors for injecting an event. First, the duration of the event, and second, the percentage of the clients within a prefix cluster that observe the event (*i.e.,* through expanded OWDs).

Our canonical "event" takes place in a single /24 prefix cluster (256 clients) where at least 20% of the constituent clients exchanged NTP packets with the server. We randomly

[2]Other NTP servers exhibited similar variance thresholds and are not shown here due to space constraints.

increase OWDs on 75% of the observed clients by a factor of 4 to 5 over a period of 7 minutes. Our first test injects an "event" into a /24 prefix cluster in the test trace every 25-minutes. Figure 2-(top) illustrates the OWDs for the synthetic events. Figure 2-(bottom) shows the RPCA scores from Tezzeract, along with the scoring distance threshold. Tezzeract methodology easily identifies all of these injected events, without any false negatives or positives.

Next, we inject a set of random events controlling both the duration and the percentage of clients that experience the event, *p*. The duration of these events was between 5 minutes to 22 hours. *p* is varied from 25 to 100%. Similar to the canonical event, we modify existing packets by increasing OWDs and add no new exchanges to the trace. To determine the increase in delay, we consider the inter-packet spacing along with polling intervals to identify the maximum possible delay. Table II shows the results of applying Tezzeract to the modified trace. While no false positives were generated, as the value of *p* decreases, the number of false negatives climbs—up to 17.9% when only 25% of the clients in the prefix experience the event. Closer examination revealed that all of these events had a duration of less than 20 minutes and thus were not detected simply due to lack of data. We ran many additional experiments with different values for event durations and percentage of clients affected, and the results were consistent with what we report here—no false positives and low false negatives.

TABLE II
Summary of injected events.

Percentage of clients in the prefix	Total number of events injected	Total number of events Tezzeract detected
100	1,000	1,000
75	1,000	983
50	1,000	962
25	1,000	821

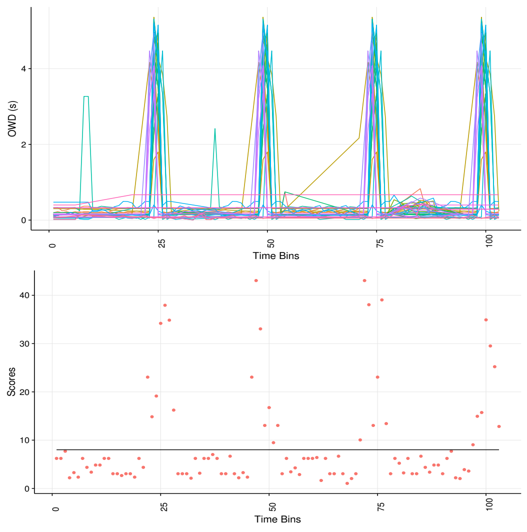

Fig. 2. **OWDs (top) and RPCA scores (bottom) for 7-minute synthetic events with 25-minute inter-event spacing on `/24` cluster**

B. Characteristics of events in NTP trace data

We apply Tezzeract to the full set of NTP trace data described in §II.

Individual Servers. Table III shows the summary of the average number of events identified by Tezzeract per day. The number of corresponding prefixes (prefix clusters) that are affected is also shown. The results show that the average number of events identified by Tezzeract varies between NTP servers: from 57k events over 6.7k prefixes (MI1) to 77 events for 8 prefixes (MW1). Comparing the number of prefixes with events (in Table III) with the total number of prefixes reported in Table I, we observe high event occurrences for university-related servers (∼98% for UI3). The results also show that the number of prefixes affected as seen from the ISP servers are fewer than the university servers. We hypothesize the lower number of events observed at the ISP-based NTP servers is due to the narrower reach of client prefixes. Many of these events are likely to be due to daily route changes, which are known to cause temporary increases in packet latencies [25].

TABLE III
Summary of events detected by Tezzeract.

Server ID	c2s		s2c	
	Avg. #events per day	#prefixes affected	Avg. #events per day	#prefixes affected
AG1	4,246	2,342	3,701	2,241
CI1	243	35	216	37
CI2	117	19	118	18
CI3	158	21	129	20
CI4	137	22	146	21
EN1	125	22	128	21
EN2	84	23	92	22
JW1	480	118	701	117
JW2	611	132	944	126
MW1	77	8	75	8
MW2	4,975	1,321	4,868	1,182
MW3	624	80	706	79
MW4	3,586	914	3,276	793
MI1	58,434	6,874	57,396	6,792
PP1	357	151	206	141
SU1	2,520	5,620	1,928	4,954
UI1	2,949	496	2,872	446
UI2	8,570	863	8,128	804
UI3	6,819	664	6,118	627

Figure 3 depicts the number of events and prefixes affected as seen from the PP1 NTP server. The figure shows that the number of events observed for the c2s path of PP1 NTP server is as high as 890; whereas it is 400 events for the s2c path. This highlights well the known issue of Internet routing asymmetries [26] and can serve as evidence for further diagnosis of the events. Interestingly, we observe a drop in the number of events and affected prefixes for the PP1 commercial NTP server over December 25 and 26, 2015. This "Christmas holiday effect" is consistent across all commercial NTP servers, but *not* consistently observed through ISP- and university-based NTP servers. We also find that the number of events detected per day varies based on whether c2s or s2c OWDs are considered once again highlighting the issue of routing asymmetries.

To complement Table III, Figure 4 shows box-and-whiskers plot of event durations for c2s events. Note that we do not show all outliers in this plot to avoid visual clutter. The figure shows that the interquartile ranges for the ISP-based NTP server (*i.e.*, SU1) are very tight. On the contrary, even though the interquartile ranges for certain servers (*e.g.*, MI1 and CI1) are comparable with other NTP servers, their maximum observed event duration exceeds 6000s (∼1.5 hours). The

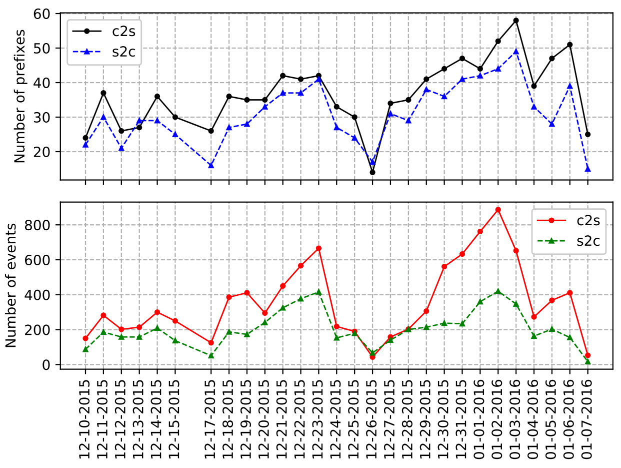

Fig. 3. **Number of events and affected prefixes for PP1**

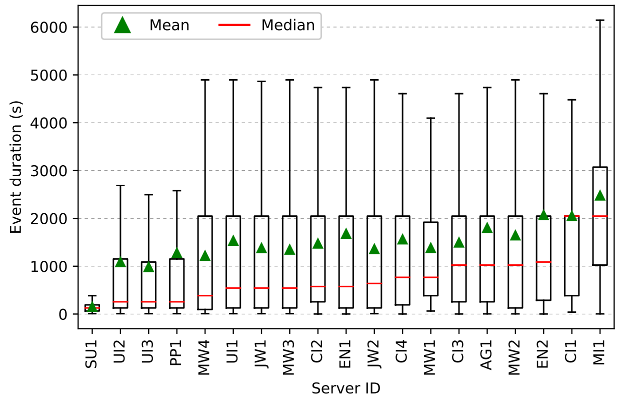

Fig. 4. **Box-and-whiskers plots showing event duration characteristics of c2s events for NTP servers.**

figure also shows that the 75th percentile of event duration for a majority of the NTP servers that we consider is ∼2000s (∼33 minutes). Finally, despite the median of event durations for a majority of servers being very low (*i.e.*, generally less than 15 minutes), we observe that the maximum event durations are fairly high. One explanation for high event duration values is clients with long NTP polling intervals (*e.g.*, 1024s), thus these values are not necessarily reflective of event duration. s2c paths also exhibited similar characteristics. Enhancing event duration estimates is a topic of future work.

Across Servers. Next, we count the unique number of prefixes and events seen across the NTP servers. Figure 5 shows the number of non-overlapping events (y axis) as observed from at least n NTP servers (x axis). All the reported counts are time-aligned *i.e.*, both events and prefixes are simultaneously identified by Tezzeract at one or more NTP servers. We observe over 11.3k and 10.8k unique prefixes from different NTP servers in c2s and s2c directions, respectively. From these prefixes, events from ∼25%, ∼5% and ∼3% of the prefixes are seen across at 2, 3, or 4 NTP servers, respectively. Figure 5 indicates events observed by as many as 10 NTP servers for certain prefixes (*e.g.*, `108.192.0.0/16` belonging to AT&T, Inc.).

Overall, these results demonstrate how NTP can offer a unique Internet-wide perspective. In particular, Figure 5 shows how prefixes could be simultaneously monitored from multiple servers. Moreover, when events are observed from multiple servers for any given prefix at the same time, they can provide a natural self-consistency check and aid in diagnosis.

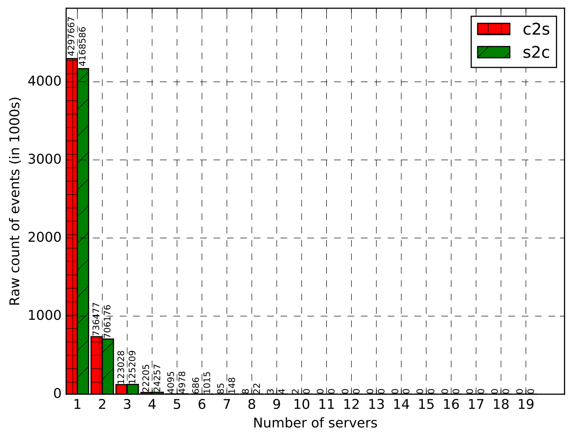

Fig. 5. **Number of events seen across multiple servers.**

C. *Consistency checking with active probe-based data*

Next, we compare and contrast the events identified by Tezzeract versus those identified by the ongoing Internet-wide Census and Survey project at ISI [12]. The ISI Census and Survey project is an example of what we consider a *core-to-edge active probing system*. Tezzeract, on the other hand, has a specific goal of identifying events based on passive observation of NTP traffic and which provides *both* core-to-edge and edge-to-core perspectives. Given the differences between the two methods and given the paucity of ground truth information to validate the identified events, the comparison of the number of events and affected prefixes identified by Tezzeract vs. census and survey should be interpreted as a way to strengthen and learn from each other. That is, Tezzeract can be used to provide a better perspective on edge-to-core events as the majority of clients behind network address translators are oblivious to active probing efforts.

Our comparison algorithm for the ISI census and survey data takes two inputs: *(1)* the ISI-identified events (*i.e.*, those recorded in /24 prefixes through active probing) and *(2)* the events identified using Tezzeract. Note that both inputs also contain the timeline of event occurrences. Given these inputs, the algorithm populates a trie with Tezzeract-identified event prefixes and their corresponding timelines (*i.e.*, event start and end times) since the events identified by Tezzeract contain prefixes larger than /24. Next, for each ISI event reported, we check whether the prefix associated with the ISI event is in the trie (or is contained within a prefix in the trie). If a match is found, the algorithm finds the best matching NTP event corresponding to an ISI event by comparing the start time of each NTP event with the ISI event. At the end of this process, we obtain a one-to-one NTP and ISI event *match list*.

Given the match list with two start and end times, say <I1, I2> from ISI and <N1, N2> from Tezzeract, the algorithm assigns the events into one of 11 categories based on the conditions given in Table IV. For categories 1 to 9, we have some form of overlap between Tezzeract- and ISI-identified events, whereas for categories 10 and 11, there are no overlaps. For categories 10 and 11, we use a pre- and post-match window of 1 hour.

TABLE IV
Categories used for comparing Tezzeract- vs. ISI-identified events.

Category	Condition
1	I1 = N1 and I2 = N2
2	I1 = N1 and I2 < N2
3	I1 = N1 and I2 > N2
4	I1 < N1 and I2 = N2
5	I1 < N1 and I2 > N2
6	I1 < N1 and I2 < N2
7	I1 > N1 and I2 = N2
8	I1 > N1 and I2 > N2
9	I1 > N1 and I2 < N2
10	I1 > N2
11	I2 < N1

TABLE V
Summary of Tezzeract events comparison with ISI events.

Server ID	OWD direction	Total # of ISI event matches with Tezzeract	Total exact /24 matches	Total C1 to C9	C10 and C11
AG1	c2s	68,895	123	48,198	20,697
	s2c	62,460	113	45,228	17,232
CI1	c2s	3,408	2	2,785	623
	s2c	2,773	1	2,317	456
CI2	c2s	1,569	0	1,185	384
	s2c	1,460	6	1,104	356
CI3	c2s	3,513	0	2,958	555
	s2c	2,409	1	2,088	321
CI4	c2s	2,132	6	1,645	487
	s2c	2,306	6	1,890	416
EN1	c2s	1,188	8	980	208
	s2c	1,198	8	994	204
EN2	c2s	807	5	619	188
	s2c	1,077	6	874	203
JW1	c2s	6,988	13	6,185	803
	s2c	8,555	16	7,487	1,068
JW2	c2s	6,657	0	5,533	1,124
	s2c	9,736	0	7,826	1,910
MW1	c2s	1,065	0	756	309
	s2c	1,053	0	725	328
MW2	c2s	79,330	290	55,548	23,782
	s2c	73,749	322	52,020	21,729
MW3	c2s	8,715	63	6,278	2,437
	s2c	10,537	69	7,406	3,131
MW4	c2s	55,406	118	38,038	17,368
	s2c	47,793	131	33,854	13,939
MI1	c2s	959,455	2,667	786,951	172,504
	s2c	951,396	2,589	784,808	166,588
PP1	c2s	3,628	3	2,733	895
	s2c	2,875	4	2,307	568
SU1	c2s	17,717	71	7,606	10,111
	s2c	14,194	47	6,346	7,848
UI1	c2s	32,477	142	23,133	9,344
	s2c	30,849	143	22,362	8,487
UI2	c2s	76,328	261	49,487	26,841
	s2c	72,151	279	48,181	23,970
UI3	c2s	59,919	151	39,339	20,580
	s2c	54,913	143	36,638	18,275

Table V shows the comparison of Tezzeract-identified events (for both c2s and s2c directions) versus events identified by census and survey. The table highlights the number of ISI events which we were able to match with Tezzeract events, the number of exact /24 matches (cases where Tezzeract's prefix cluster is a /24, thereby enabling a direct one-on-one comparison with the ISI outage event), and the number of events under various categories. From this table, we see that as much as ∼67% of the events identified by Tezzeract from the CI2 NTP server match with those identified by census and survey. We further observe that on average, over 66% of the Tezzeract-identified events overlap in terms of event timelines across all the servers (in C1 to C9 category); the remaining (∼33%) of Tezzeract-identified events occur within a match window of one hour, either *before* or *after*, those identified by ISI (see C10 and C11). Next, we see that there are more /24 exact matches between the Tezzeract and ISI for commercial- and university-based NTP servers in comparison with the ISP counterparts. This may simply reflect the fact that NTP clients that contact the ISP servers typically come from larger routing prefix aggregates. Lastly, we observe that Tezzeract finds ∼63% new events, on average, across all the NTP servers considered in this study. This is likely due to the reach of NTP clients vs. the probing cycle of the ISI servers.

The unique events that were identified by Tezzeract consisted of both network prefixes covered as part of the ISI survey, as well as prefixes which were not reachable via standard active probing methods. The median duration of these events is approximately 20 minutes, which is similar to the overall median duration. Tezzeract was able to identify many long duration events, observed by multiple clients within the affected prefix clusters. The top two longest duration events observed lasted for approximately 2 and half hours and affected the following prefixes: `204.93.0.0/19` which belongs to AS 698 (University of Illinois) and `54.186.0.0/15` which belongs to AS 16509 (Amazon.com, Inc.). These events were observed by 53 and 74 unique clients belonging to the corresponding prefix clusters. These events are likely to be outages similar to those discussed below.

D. Consistency checking with other sources

Finally, we compare events identified by Tezzeract with public reports. In particular, we rely on the Outages mailing list [16] to further enhance the confidence in the events identified by Tezzeract. To illustrate with an example, we consider the outage event [27] discussed by Level3's (now CenturyLink) administrators in the Outages mailing list as ground truth. On December 15, 2015 an outage event occurred because of a router addition that impacted multiple users and businesses between Chicago, IL and Atlanta, GA.

To evaluate whether the events identified by Tezzeract coincide with this known outage, we first selected all the events identified on December 15, 2015, resulting in 148,322 events. Similar to our comparison with ISI event data, we use four timestamps: *(1)* two from each event (start and end) identified by Tezzeract, and *(2)* start and end timestamps derived from the Outages mailing list [27], which are December 15, 2015 18:18 GMT and December 15, 2015 19:20 GMT. Next, we find the time-alignment category under which the Tezzeract-identified events fall with respect to [27] (see Table IV). If the events fall under C1 to C9, we note those events; otherwise, we ignore those events. For the events noted, we extract the list of client IP addresses and their corresponding geographic coordinates using MaxMind's IP Geolocation service [28]. Subsequently, we match the geographic locations from [16] and the ones obtained above.

From the 148k events initially identified, the process outlined above results in 1,104 events overlap with the ground

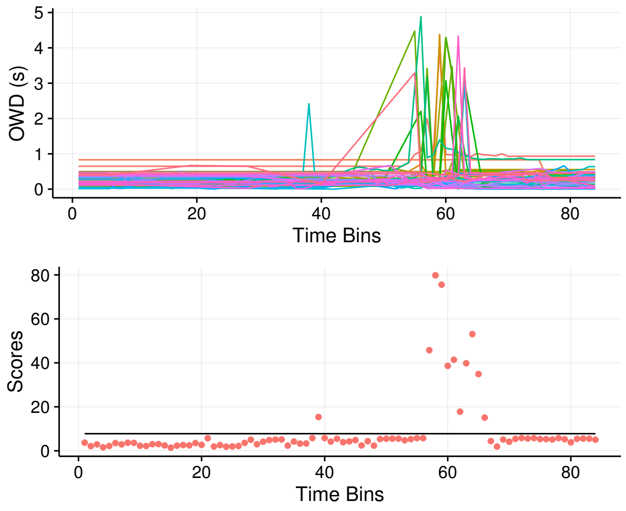

Fig. 6. **Level3 outage event identified by Tezzeract affecting AS 32748.**

truth [27] in terms of event timelines, prefixes and geographic locations. Figure 6 shows the OWD spikes (top) and scores for clients (bottom) in one of the prefixes affected. The OWD spikes represented in Figure 6-(top) contain 43 unique clients from the prefix `208.117.0.0/18` which belongs to AS 32748 (Steadfast.net), a peer of Level3. The results depicted in this plot highlight the effectiveness of Tezzeract in identifying Internet outages. In particular, we observe that the effect on NTP-derived OWDs is extreme for clients within each of these ASes, underscoring the severity of the event. Tezzeract can also offer an alternate perspective to BGP-based event detection tools [29][3].

V. Related Work

Prior research on network outages has considered various perspectives including rerouting or routing anomalies caused by failures in *core and transit networks*, connectivity outages and performance impairments for customers in the *network edge*, and *service outages*.

Core and transit network outages. Core network failures and performance anomalies have been examined using both passively collected data sources and active measurement. Analysis of inter-domain (BGP) and intra-domain (*e.g.*, IS-IS, OSPF, etc.) routing updates have formed the basis of many of the studies based on passive data collection, *e.g.*, [30]–[32]. In a related vein, Banerjee *et al.* used the outages mailing list [16] as the basis for evaluating core network failures [33]. Yet another source of passive data for detecting and analyzing wide-area faults has been through analysis of background radiation traffic [34]. Active measurement techniques have also been widely used to detect routing loops and other anomalies and path failures [35]. Tomographic techniques have also been developed to actively probe a network in order to detect faults and localize them to particular links or subpaths [36].

Outages at the network edge. Active measurement has been the dominant technique for detecting failures at the edge of the network. Periodic pings combined with analysis of BGP updates were used to trigger traceroutes to verify and monitor edge network outages in Hubble [37]. The study and ongoing data collection by Quan *et al.* employ low-rate pings to the entire IPv4 address space, and detect outages and disruption events through a Bayesian formulation [4]. In our study, we use a subset of these data for comparison and validation. In their study, Padmanabhan *et al.* study the response time to pings across the IPv4 Internet and find that 5% of responses from 5% of addresses take at least 5 seconds to arrive [38]. This finding has important implications for the design of any active measurement-based system that uses the lack of responses to identify outages. Our event detection strategy is based on *passively* collected NTP data.

PCA-based analysis of network data. Identifying events of interest from streams of Internet data using PCA has been of interest to the measurement community for many years. This includes applying PCA to detect BGP anomalies [39], network traffic anomaly detection and diagnosis [2], [40], network monitoring and anomaly detection [41], and network diagnosis [42]. Issues with PCA-based methods are pointed out by [43], [44]. Similarly, sensitivity of PCA to calibration and its corresponding implications to anomaly detection are discussed by Ringberg *et al.* [20]. Our event detection approach applies Robust PCA, instead of PCA, on NTP traces and demonstrates a new, unique perspective for Internet-level event detection without additional infrastructure.

VI. Summary and Future Work

Understanding the scope and nature of unexpected network events is important for effective management and operation of communication networks. In this paper, we address the problem of Internet event detection by developing a novel framework and implementation called Tezzeract. Our framework uses one way delays (OWDs) extracted from Network Time Protocol (NTP) packet exchanges and therefore does not require any new infrastructure to be deployed, or any additional network traffic and offers the opportunity for a broad perspective on events. Our algorithm for identifying events is based on Robust PCA, which is resilient to noisy data that is typical in the Internet measurements including NTP data. Our implementation of Tezzeract produces a characterization of events including the number of NTP clients affected, event duration, and prefix(es) affected.

We assess Tezzeract in a series of controlled experiments based on injecting synthetic events in an NTP trace. We find that Tezzeract is highly accurate in reporting with a false negative rate related to measurement reach. We then apply Tezzeract to a large NTP data set collected from 19 servers in the US. We find that the average number of events per day varies widely across the set of servers, as do the event durations. We observe that the median event duration for most servers is approximately 20 minutes or less, but that the distribution is skewed, with quite a few very long events (*e.g.*, more than 1 hour). We also find that a considerable number of events are observed at more than 1 NTP server, *e.g.*, 25% of events are observed at 2 servers, and that we observe

[3]These results are not shown here due to space constraints.

some events at 10 of the 19 servers indicating significant impact across clients. We compare the events detected through Tezzeract with event data collected through the ongoing ISI census and survey project and find that between 21–67% of events that are detected by Tezzeract are also identified by the ISI system, but that Tezzeract also identifies new events. We also evaluate events detected by Tezzeract in relation to a reported outage. We observe that the event is identified through different sets of NTP clients on different prefixes, and but that the different client populations are impacted in similar ways.

In ongoing work, we are examining additional ways to corroborate and gain perspective on the events detected by Tezzeract, and are exploring ways to automate this process. We are also examining events recognized by *decreases* in OWDs, which is a simple extension of Tezzeract. Finally, we are also examining the possibility of performing real-time event detection and analysis.

Acknowledgments

We thank the NTP operators for providing the datasets. This work is supported by NSF grants CNS-1703592, DHS BAA 11-01, AFRL FA8750-12-2-0328. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, DHS, AFRL or the U.S. Government.

References

[1] "Effects of Catastrophic Events on Transportation System Management and Operations: Howard Street Tunnel Fire Baltimore City, Maryland," https://ntl.bts.gov/lib/jpodocs/repts_te/13754.html.

[2] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies Using Traffic Feature Distributions," *ACM SIGCOMM*, 2005.

[3] P. Barford, N. Duffield, A. Ron, and J. Sommers, "Network performance anomaly detection and localization," in *IEEE INFOCOM*, 2009.

[4] L. Quan, J. Heidemann, and Y. Pradkin, "Trinocular: Understanding Internet reliability through adaptive probing," in *ACM SIGCOMM Computer Communication Review*, 2013.

[5] B. Eriksson, R. Durairajan, and P. Barford, "Riskroute: A framework for mitigating network outage threats," in *ACM CoNEXT*, 2013.

[6] M. Zhang, C. Zhang, V. S. Pai, L. L. Peterson, and R. Y. Wang, "Planetseer: Internet path failure monitoring and characterization in wide-area services." in *OSDI*, 2004.

[7] Y. Zhu, B. Helsley, J. Rexford, A. Siganporia, and S. Srinivasan, "Latlong: Diagnosing wide-area latency changes for cdns," *IEEE Transactions on Network and Service Management*, 2012.

[8] R. Durairajan and S. Mani and J. Sommers and P. Barford, "Time's Forgotten: Using NTP to Understand Internet Latency," in *ACM HotNets*, 2015.

[9] R. Durairajan, S. K. Mani, P. Barford, R. Nowak, and J. Sommers, "TimeWeaver: Opportunistic One Way Delay Measurement via NTP," https://arxiv.org/abs/1801.02123, 2018.

[10] N. Locantore, J. Marron, D. Simpson, N. Tripoli, J. Zhang, K. Cohen, G. Boente, R. Fraiman, B. Brumback, C. Croux *et al.*, "Robust Principal Component Analysis for Functional Data," *Test*, 1999.

[11] M. Hubert, P. Rousseeuw, and T. Verdonck, "Robust pca for skewed data and its outlier map," *Computational Statistics & Data Analysis*, 2009.

[12] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, "Census and Survey of the Visible Internet (extended)," *ISI-TR-2008-649*, 2008.

[13] "CAIDA Routeviews Prefix to AS mappings Dataset (pfx2as) for IPv4 and IPv6." https://www.caida.org/data/routing/routeviews-prefix2as.xml.

[14] "Team CYMRU's IP to ASN mapping." http://www.team-cymru.org/IP-ASN-mapping.html.

[15] "USC/LANDER Project. Internet Outage Dataset, PREDICT ID: USC-LANDER/internet_outage_adaptive_a22all-20151001 and USC-LANDER/internet_outage_adaptive_a23all-20161001," http://www.isi.edu/ant/lander.

[16] "Outages Mailing List." https://puck.nether.net/mailman/listinfo/outages.

[17] C. Croux and G. Haesbroeck, "Principal Component Analysis based on Robust Estimators of the Covariance or Correlation matrix: Influence Functions and Efficiencies," *Biometrika*, 2000.

[18] M. Hubert and P. J. Rousseeuw and K. Vanden Branden, "ROBPCA: A New Approach to Robust Principal Component Analysis," *Technometrics*, 2005.

[19] S. Serneels and T. Verdonck, "Principal Component Analysis for Data Containing Outliers and Missing Elements," *Computational Statistics and Data Analysis*, 2008.

[20] H. Ringberg and A. Soule and J. Rexford and C. Diot, "Sensitivity of PCA for Traffic Anomaly Detection," *SIGMETRICS*, 2007.

[21] "R package: Classical Or Robust Principal Components For Incomplete Data." https://www.rdocumentation.org/packages/rrcovNA/versions/0.4-8/topics/PcaNA.

[22] C. Croux and A. Ruiz-Gazen, "High Breakdown Estimators for Principal Components: The Projection-Pursuit Approach Revisited," *Journal of Multivariate Analysis*, 2005.

[23] C. Croux, P. Filzmoser, and M. R. Oliveira, "Algorithms for Projection–Pursuit Robust Principal Component Analysis," *Chemometrics and Intelligent Laboratory Systems*, 2007.

[24] C. Pascoal, M. R. De Oliveira, R. Valadas, P. Filzmoser, P. Salvador, and A. Pacheco, "Robust feature selection and robust pca for internet traffic anomaly detection," in *IEEE INFOCOM*, 2012.

[25] H. Pucha, Y. Zhang, Z. M. Mao, and Y. C. Hu, "Understanding network delay changes caused by routing events," in *ACM SIGMETRICS Performance Evaluation Review*. ACM, 2007.

[26] A. Pathak, H. Pucha, Y. Zhang, Y. C. Hu, and Z. M. Mao, "A Measurement Study of Internet Delay Asymmetry," in *PAM*, 2008.

[27] "[outages] Level3 Chicago?" https://puck.nether.net/pipermail/outages/2015-December/008572.html.

[28] "MaxMind IP Geolocation Service." https://www.maxmind.com/.

[29] M. Syamkumar, R. Durairajan, and P. Barford, "Bigfoot: A Geo-based Visualization Methodology for Detecting BGP Threats," in *IEEE Symposium on Visualization for Cyber Security*, 2016.

[30] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," *IEEE/ACM Transactions on Networking*, 1998.

[31] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of Internet stability and backbone failures," in *International Symposium on Fault-Tolerant Computing*, 1999.

[32] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot, "Characterization of failures in an operational ip backbone network," *IEEE/ACM Transactions on Networking*, 2008.

[33] R. Banerjee, A. Razaghpanah, L. Chiang, A. Mishra, V. Sekar, Y. Choi, and P. Gill, "Internet outages, the eyewitness accounts: Analysis of the outages mailing list," in *PAM*, 2015.

[34] K. Benson, A. Dainotti, K. C. Claffy, and E. Aben, "Gaining insight into AS-level outages through analysis of Internet background radiation," in *IEEE INFOCOM workshops*, 2013.

[35] V. Paxson, "End-to-end routing behavior in the Internet," *IEEE/ACM transactions on Networking*, 1997.

[36] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data," in *ACM CoNEXT conference*, 2007.

[37] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. E. Anderson, "Studying Black Holes in the Internet with Hubble," in *NSDI*, 2008.

[38] R. Padmanabhan, P. Owen, A. Schulman, and N. Spring, "Timeouts: Beware surprisingly high delay," in *ACM IMC*, 2015.

[39] K. Xu and J. Chandrashekar and Z.L. Zhang, "A First Step toward Understanding Inter-domain Routing Dynamics," in *ACM SIGCOMM workshop on Mining network data*, 2005.

[40] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-wide Traffic Anomalies," in *ACM SIGCOMM*, 2004.

[41] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, and G. Maciá-Fernández, "PCA-based Multivariate Statistical Network Monitoring for Anomaly Detection," *Computers & Security*, 2016.

[42] X. Li, F. Bian, H. Zhang, C. Diot, R. Govindan, W. Hong, and G. Iannaccone, "MIND: A Distributed Multi-Dimensional Indexing System for Network Diagnosis," in *IEEE INFOCOM*, 2006.

[43] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural computation*, 1998.

[44] M.E. Tipping and C.M. Bishop, "Mixtures of Probabilistic Principal Component Analyzers," *Neural computation*, 1999.