Disco: Fast, Good, and Cheap Outage Detection

Anant Shah*, Romain Fontugne†, Emile Aben§, Cristel Pelsser‡, and Randy Bush†
*Colorado State University, †IIJ Research Lab, ‡Université de Strasbourg, §RIPE NCC
akshah@cs.colostate.edu, {romain,randy}@iij.ad.jp, emile.aben@ripe.net, pelsser@unistra.fr

Abstract—**Outage detection has been studied from different angles, such as active probing, analysis of background radiations, or control plane information. We approach outage detection from a new perspective. `Disco` is a detection technique that uses existing long-running TCP connections to identify bursts of disconnections. The benefits are considerable as we can monitor, without adding a single packet to the traffic, Internet-wide swaths of infrastructure that were not monitored previously because they are, for example, not responsive to ICMP probes or behind NATs. With `Disco` we analyze state changes on connections between RIPE Atlas probes and the RIPE Atlas infrastructure. This data, that is originally logged to monitor probe availability, has a small footprint and is available as a publicly accessible live stream, which makes light-weight near real-time outage detection possible. Probes perform planned traceroute measurements regardless of their connectivity to the RIPE Atlas infrastructure. This gives us a no cost advantage of viewing the outage inside out as the probes experienced it, characterizing the outage after the fact. Thus, we present an outage detection system able to run in near real-time (fast), with a precision of 95% (good), and without generating any new measurement traffic (cheap). We studied historical probe disconnections from 2011 to 2016 and report on the 443 most prominent outages. To validate our results we inspected traceroute results from affected probes and compared our detection to that of Trinocular.**

I. Introduction

Internet outages pose a challenge for the day-to-day operations of networks supporting millions of users and businesses. Many research projects are throwing millions of packets all over the Internet with the goal of detecting connectivity losses. This project mines pre-existing data, generating no new measurement packets, in order to measure connectivity issues.

The technique, which we call `Disco`[1], is not restricted to parts of the Internet which respond to active probing, as it uses stable long-running TCP connections from wide-spread infrastructure, much of which is behind NATs and other measurement blockage. A single TCP disconnect can be ascribed to very local effects on either end of the connection, but a burst of TCP disconnects is a tell-tale sign of a more serious outage. We define an outage as bursts of disconnects from a certain geography or topology.

To detect bursts of disconnects we developed a modified version of Kleinberg's burst model based on a probabilistic infinite state automaton [18]. Unlike simple threshold techniques based on disconnect counts, this approach effectively models the temporal dynamics of the analyzed data and accurately reports disconnect bursts [19]. Kleinberg's burst detection model has attracted attention from various research domains including human behavior analysis [7], social network analysis [15], and web analysis [20].

We apply `Disco` to the stable long-running TCP connections between the RIPE controlling infrastructure and all Atlas probes deployed in about 3.3k Autonomous Systems (ASs). A disconnection is recorded by the controlling infrastructure when a probe looses connectivity. This log of disconnections, available both as historical data and live stream, is the input to `Disco`. Probes are typically well deployed in large residential ISPs in Europe and North America. These networks are best monitored with the combination of RIPE Atlas and `Disco` as they are usually behind NATs and do not respond to active probing. Therefore, visibility provided by the proposed approach differs from popular community outage detection system Trinocular [21]. Out of the total Internet space visible to RIPE Atlas, about 25% is not monitored by Trinocular.

Atlas disconnection data is available in near real-time, and can be used to detect network issues very shortly (in the order of a minute) after they occur. Also, when a probe loses connectivity to the controlling infrastructure, it keeps running planned traceroute measurements, enabling more fine grained analysis of the cause. We also leverage pre-existing information about the probes' AS, country and geolocation to gain high detection accuracy and provide more insights about the outages.

This unique combination of data and model makes `Disco`:

Fast: By use of our modified version of Kleinberg's burst model, we were able to process 6 years worth of data in only 103 minutes on a single machine. It also enables to run live outage detection on the Atlas data stream.

Good: By use of multi-resolution analysis, namely aggregating probes by their topological or geographical similarities, `Disco` precisely locates outages in time and space. It reports relevant outages while missing only a few (precision of 95% and recall of 67%).

Cheap: By use of TCP connection status data `Disco` is not generating any new measurement packet to detect outages. It also uses pre-existing traceroutes to characterize the outage after the fact. No additional measurements were scheduled, nor additional logs generated for this work.

In the remainder of this paper we first present related work (Section II) and then our datasets (Section III). Next we describe how we detect bursts (Section IV). We investigate the traceroutes run on the probes during the outage and compare to Trinocular in Section V. In Section VI we characterize detected outages. In Section VII we show example case studies

[1] https://github.com/romain-fontugne/disco

of select outages and elaborate more on the usability of our method and finally we conclude in Section VIII.

II. Related Work

Outage detection has been studied from different angles. Operationally, important outages are likely to be discussed on network operations mailing lists which can be data-mined [6]. Censorship can be implemented as country-wide Internet outages, which have been studied from BGP, traceroute, and Internet background radiation data in [11].

For parts of the Internet that generate enough background radiation, network telescopes [4] can be used to detect outages. The alternative is sending probing packets and detecting changes in responses. To do this at Internet scale one needs to inject a massive number of packets in the network, and this works only for the part of the Internet where targets respond to active probing and packets are not blocked by upstream infrastructure. Dainotti et. al. [10] find that, out of 10.5M routed /24 prefixes equivalents, IP addresses in 3.1M /24 prefixes are seen in the UCSD network telescope, and 4.5M /24 prefixes are visible using active probing, but both techniques observe mostly the same /24 prefixes. Also, outage detection methods based on network telescope data are opportunistic, as they are restrained to monitor whatever networks generating background radiation. Our study relies on a pre-existing metadata stream which is deterministic thus more reliable. Furthermore, we observed that out of the total Internet space visible to RIPE Atlas, about 25% is not monitored by active probing.

In [26], an approach is proposed solely based on flow data at a network border. To make this work across multiple networks, the problem of sharing potentially privacy sensitive flow data must be tackled, which is explored in privacy preserving distributed outage monitoring [12], [13], with non-trivial complexity.

In active probing, the Trinocular [21] and Hubble [17] projects are especially notable, as they do Internet-wide adaptive scanning. Trinocular uses adaptive ICMP probing, exploring the trade-off between probing rate and accuracy. Hubble uses BGP feeds and ICMP probing to find potential problems which are investigated and classified using traceroute-based approaches. PlanetSeer [27] detects anomalies in P2P traffic. Upon anomaly detection it performs active probing to locate and quantify the outage.

III. Data Sources

We use separate data sources for detection, validation, and classification of outages as shown in Table I. All data we use is public and pre-exists, meaning that it is gathered for other prior purposes to our work.

A. For Outage Detection

To detect outages we use the RIPE Atlas infrastructure [23], which is a global deployment of "probes" (tiny Linux machines) that continuously perform measurements such as pings and traceroutes. While there were more than 9,300 probes active in 178 countries on October 14, 2016, many more probes came and went during the study.

Usage	Name	Sources
Outage Detection	Probe Metadata	RIPE Atlas
Outage Validation	a) Pings b) Traceroutes	a) Trinocular (USC/ISI) b) RIPE Atlas
Outage Characterization	Traceroutes	RIPE Atlas

TABLE I: Dataset Description

RIPE Atlas probes receive measurement commands and send measurement results back via SSH connections over TCP port 443 to a set of servers called "controllers". At probe boot, a connection is established to a single controller, which records the connect event. SSH keep-alives are used to check if both sides of the connection are up. If a controller finds a probe unresponsive for more than a minute, it tears down the TCP connection and records a disconnect event for that probe. The set of dis/connect events is available through the RIPE Atlas API both as an historic dataset [24] and a real-time stream [25], and is the sole input of our outage detector. In this paper, we use data from 2011 to 2016 to detect bursts of disconnects and diagnose outages.

B. For Validation and Characterization of Outages

For characterization we use traceroutes from RIPE Atlas probes. For validation we use the same traceroute dataset as well as pings from Trinocular [21].

RIPE Atlas probes frequently run traceroute measurements. When probes are not connected to controllers, they buffer the traceroute results for up to six hours and send it to a controller on the next successful connection. We take advantage of this to compare traceroutes before and during the detected outages. If the probe indeed experienced an outage, the buffered traceroute would not be complete i.e., would not reach the targeted destination during that time.

The periodic traceroutes we rely on are the "built-in" and "anchoring" measurements. These are available as public data to anyone. Built-ins are traceroutes to DNS root servers (mostly anycast) while anchoring measurements are traceroutes to Atlas anchors located in various parts of the world. We use these traceroutes to validate that detected bursts of disconnects correspond effectively to outages (Section V) and to characterize the detected outages (Section VI).

The second dataset we use for validation is that of the Trinocular project. From four geographically diverse vantage points, Trinocular pings four million /24s to track responsive blocks. Based on the responses Trinocular infers the state of a /24 prefix as up, down or uncertain using the methods explained in [21]. In particular, we used the processed 'outage adaptive datasets' [5] from April 2015 to December 2015. This gives us an external source of information to validate the unavailability of prefixes. However, out of the 10.5K /24s where RIPE Atlas probes are located, only 7.9K /24s are monitored by Trinocular and 2.6K /24s are not. Using RIPE

Atlas probes for outage detection, we monitor this previously unseen address space without generating any probing traffic.

IV. Outage Detection

We process disconnection data in three main steps: (1) model the arrival rate of probe disconnects (2) perform burst detection on sub-streams (a collection of probes that share geographical or topological characteristics) of data and (3) report significant events. The next subsections explain each in detail.

A. Burst Modeling

The significance of a burst is characterized by the number of disconnects and their arrival rate. The temporal characteristics of bursts are poorly modeled by simple time series of the number of disconnects. Indeed counting disconnects in time bins conceals the exact temporal relations between consecutive disconnects. Disconnects that are uniformly spread out through a time bin should be considered differently than synchronous disconnects. For example, if we use a one minute time bin to count disconnect events, we will treat similarly three disconnects that occur within the same second and three disconnects that are uniformly spread out through the time bin. In our study, however, we want to put more emphasis on the three disconnects that happened at the same second, since, synchronous disconnects are a stronger sign of outages. Of course a smaller time bin would help in this particular case, however, the time series does not provide insights on the distribution of the disconnections. Moreover, excessively small time bins would overly fragment related disconnects across several time bins. Since such methods are very sensitive to the bin size, applying them on variety of data sources is very challenging.

To alleviate this time scale issue, we model the disconnects arrival rate with the infinite-state automaton proposed by Kleinberg [18]. States represent different arrival rates of disconnect events, and an algorithm is devised to find the state sequence corresponding to a stream of disconnections.

Formally, disconnections for a period of time T are represented as a stream of inter-arrival times $x = (x_1, x_2, ..., x_n)$, so that x_1 is the time spent between the first and the second observed disconnection. Each state q_j in the automaton is associated with an exponential density function $f_j(x_t) = \alpha_j e^{-\alpha_j x_t}$ with rate $\alpha_j = \frac{n}{T} 2^j$. Therefore, the arrival rate corresponding to each state is exponentially increasing, and with $0 \leq i < j$, the state q_j represents higher intensity bursts than the ones modeled by q_i.

Kleinberg [18] proposes to find the optimal state sequence corresponding to a given stream by adapting the Viterbi algorithm [22] to the following model and cost function. $C_j(t)$ is the minimum cost of a state sequence ending with q_j for the input $x_1, x_2, x_3, ..., x_t$ and is defined by the recurrence relation:

$$C_j(t) = -\ln f_j(x_t) + \min_l (C_l(t-1) + \tau(l, j))$$

with initial conditions $C_0(0) = 0$ and $C_j(0) = \infty$ for $j > 0$, and τ the cost of transitions between states which is positive for transitions to higher states but null on return to lower states (see [18] for more details).

As explained in the next section, we separate the global stream of disconnects by country, AS and geolocation into multiple sub-streams and expect state q_j, hereafter referred as burst level j, to represent the same disconnection rate for every stream. However, with Kleinberg's original formulation comparing results obtained from different streams is difficult. The rates α associated to burst levels are function of the number of disconnections and time duration of the stream (respectively n and T). Therefore, burst levels represent different arrival rates due to different number of probes in the set. We modify the original burst model in order to calibrate burst levels based on the number of probes in a sub-stream, so that burst levels represent arrival rates per probe and burst levels computed with different sub-stream, or dates, are comparable.

Formally, the ratio $\frac{n}{T}$ is originally used to calibrate burst levels based on the analyzed stream, so that burst level 0 represents uniformly distributed disconnects (i.e. the lowest arrival rate for the analyzed stream $\alpha_0 = \frac{n}{T} 2^0$). By selecting $\frac{n}{T}$ based on the number of probes we obtain uniform burst levels across all sub-streams. The variable T is set to one day, consequently, in our experiments all analyzed streams last one day. The variable n is set to the daily number of disconnections expected for the analyzed sub-stream. As we found about one daily disconnect per RIPE Atlas probe, we set n to the number of active probes for the analyzed sub-stream. Thereby, burst levels are stable even if the number of active probes (consequently the number of disconnects) for a certain sub-stream increases from day to day.

Given a sequence of disconnect timestamps, the above algorithm produces a sequence of burst levels such as the ones in Figure 1(c) and (e). Each level represents the disconnect arrival rate normalized by the number of probes in the analyzed stream. High burst levels indicate points in time where the rate of disconnects has greatly increased, hence helping to identify outages.

As we rely on the status of TCP connections to detect network outages and there are a myriad of causes of end-point failures, the analyzed streams are particularly noisy. Our burst model is, however, inherently robust to such noise as the cost of transitions to upper states increases exponentially.

Figure 1 illustrates the number of connected probes along with the corresponding burst levels for June 7th 2016. Two known events have happened on this day, a large Kenyan power outage occurred at about 08:30 UTC, and most of the RIPE Atlas controllers have been rebooted from 13:00 UTC. The burst levels computed with all probes (Figure 1(c)) exhibits clearly the controllers reboot, but not the Kenyan outage. Instead we found bursts that appear every two hours throughout the entire day. Our manual inspection of these periodic bursts revealed times of intense measurements on v1 (old firmware) probes causing the reboot of the highly loaded probes. These bursts correspond to meaningful events affecting the RIPE Atlas platform itself that we have reported and was acknowledged by RIPE NCC. Since the number of impacted

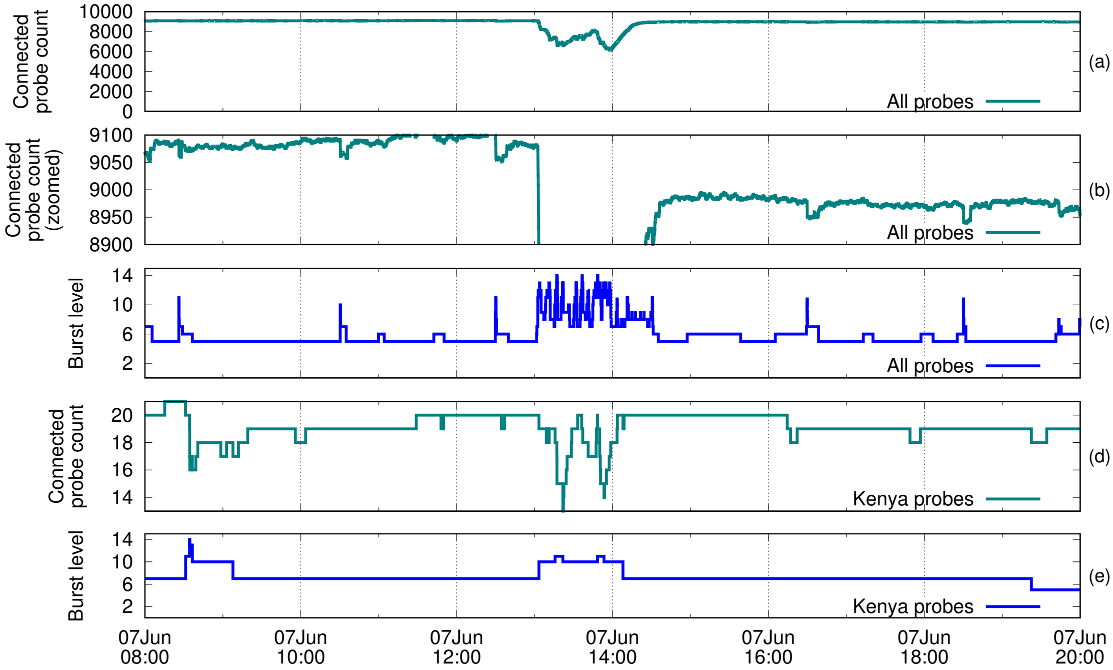

Fig. 1: Example showing raw connected probe counts and using country sub-stream of results obtained from the burst model with disconnect events reported on June 7th, 2016. Thanks to our modification of Kleinberg's burst model, burst levels with the entire stream and the Kenyan sub-stream have similar base burst level, but the relevant Kenyan events achieve higher bursts in the sub-stream.

probes is relatively small, these synchronized disconnections are particularly hard to identify from the count of connected probes. This example illustrates the benefits of the temporal analysis of our burst model.

Due to the low number of probes in Kenya, neither a drop in probe count (Figure 1(a) and (b)) nor a burst is visible for the Kenyan outage (Figure 1(c)). Note that the first significant burst around 08:30 UTC in Figure 1(c) is from the v1 probes rebooting not the Kenyan power outage. By looking at data only for probes in Kenya (raw data shown in Figure 1(d)), the power outage is easily identifiable through a burst of level 14 (Figure 1(e)). In addition, by using this *'sub-stream'* the controller reboots are less emphasized because Kenyan probes are connected to different controllers hence have been disconnected asynchronously. This example illustrates again the benefits of the burst model. Although there were more Kenyan probes disconnected during controller reboot a higher burst level is reached during the power outage as probes were synchronously powered down.

B. Sub Streams

It is evident from the previous example that splitting the main stream of disconnections into *sub-streams* helps to mitigate the disparate probe deployment. A sub-stream is a collection of probes that share one of the following geographical or topological characteristic: the same country, the same AS or the same geographical area (within 50km radius). Therefore, a single probe appears in three sub-streams, a country, AS, and geoProximate sub-stream.

We justify the choice of these sub-streams as follows:
a) *country sub-streams*: The Atlas probes are distributed among different countries disproportionately. By aggregating probes that belong to the same country and analyzing them separately allows us to detect impairments in a country even if it has deployed a small number of probes.
b) *AS sub-streams*: An AS sub-stream enables us to focus on a certain part of the network topology. This is particularly interesting to network operators aiming at monitoring connectivity of their ASs even across large geographical area.
c) *geoProximate sub-streams*: The goal of this sub-stream is to detect bursts of disconnections that may belong to different ASs or even countries but are geographically close to each other. This enables us to pinpoint the geographical effect of an outage. We believe outages due to natural calamities or power outages where entire cities loose connectivity stand out here. We chose to cluster probes that are within a 50km radius. Past research has suggested using 40km to 50km radius as a city-level assumption [16].
Depending on which type of sub-stream an outage is seen, we can identify additional information about the outage. For example, in a recent power outage in Amsterdam [9], many probes belonging to different ASs suffered disconnection. This was caught by geoProximate sub-stream. Similarly, we found cases where probes of same AS, but far away geographically, suffered an outage. These were caught by country and AS sub-streams. We revisit these examples, that emphasize advantages of sub-streams, in Section VII.

C. *Outage Reporting*

Once burst model is applied on multiple sub-streams we detect large events by applying a threshold and then aggregating events that appeared in different sub-streams but can be part of the same event. Next, we elaborate on each.

Burst Threshold: To systematically distinguish significant bursts from disconnects caused by other non relevant events, we consider only bursts that reach a certain level. Manual inspection of results obtained with the modified burst model shows the burst level usually fluctuates between 4 and 8, and goes beyond 10 during outages. In Figure 2 we show the number of detected events using different threshold values. We observe a significant drop in the number of detected events from threshold 8 to 10. Higher threshold values permit to detect the most significant events at the price of missing small outages. We recommend to use threshold values between 10 and 14. In the rest of this paper we set this threshold to 12 as a mid value to keep a justifiable trade off. The precision and recall using this threshold are discussed in Section V.

A burst of disconnects reveals the start of an outage but its time duration is unknown. To estimate the end of an outage, we select all probes disconnected during the detected burst and retrieve their corresponding re-connect events. Intuitively, the first reconnect signals the end of the outage, but some probes might be wrongly accounted for as they inadvertently get disconnected during the outage. We assume that at least 50% of the probes involved in the burst are affected by the outage, consequently, we define the end of the outage as the median re-connect timestamp of the disconnected probes.

Aggregation: The final step of outage reporting is aggregation. As described previously we split disconnections by country, AS or geolocation. Therefore, a burst might appear in multiple sub-streams. For example, during an outage in *AS1* which is located in country *C*, probe disconnect bursts will appear in both sub-streams, *AS1* and *C*. We group detected outages that appear in different sub-streams but start within the same 30 minute window and end within the same 30 minute window. We discard events that start and end within the same 30 minute window, hence deliberately discarding transient events such as a controller reboot. According to information obtained from RIPE NCC the reboots are usually in the order of few minutes. Atlas controllers are updated and rebooted once in a while, breaking numerous TCP connections at once. For our study we empirically chose a minimum outage duration (30 minutes) much larger than the time observed for controller reboots, thus avoiding to report these irrelevant events. This would not be necessary if the planned events were announced in the stream of data.

D. *Detection Results*

After detecting burst events in the raw metadata of disconnect and connect events in the AS, country, and geoProximate sub-streams we perform aggregation as described above. This gave us 443 large outages between 2011 to 2016. We detected outages due to faulty maintenance issues that gained press attention: The Time Warner Cable outage on August 27th 2014 [3], the AMS-IX outage [1] and Kenyan power failure [8]. We also detected recurring outages in Benin and Andorra which were not in the limelight. We validate these results in the next section and further characterize these 443 outages in Section VI.

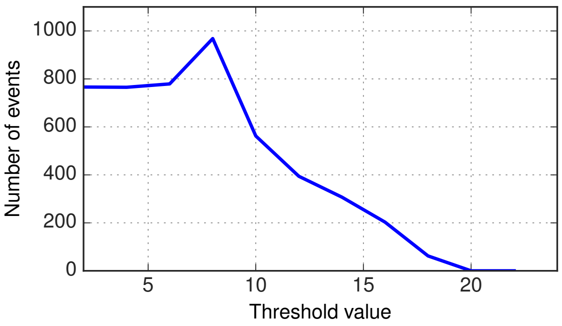

Fig. 2: Number of reported events for different threshold values.

Performance: The proposed burst model has a time complexity of $\mathcal{O}(ns^2)$ where n is the number of disconnect events and s is the number of states in the implemented automaton. Our implementation (python2.7) takes 103 minutes to analyze all types of sub-stream for 6 years of historical disconnection data from RIPE Atlas. This execution was done on Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz with 24 processes in parallel. Each process takes less than 500MB of RAM. It can easily run on the live feed of disconnection data and raise alarms for events within a few minutes of it's occurrence.

Applicability: In this paper we study TCP connections between the RIPE infrastructure and all Atlas probes deployed in 3.3k ASs. The scope of our study is hence limited to the Atlas platform deployment, but `Disco` could be deployed in larger infrastructures with long-running TCP connections, for instance large scale video platforms.

V. Validation

We validate the results of our outage detector using two distinct data sources: traceroutes from the probes involved in the disconnect bursts and Trinocular [21] ping data. The first dataset is used to check whether traceroutes sourced by the affected probes are also affected. The second dataset has the advantage of being completely independent from our experiment's infrastructure.

A. *Comparison to Traceroutes*

To measure the impact of outages with traceroutes, we define the velocity of an Atlas probe as the number of complete traceroutes (i.e. traceroutes that reached the intended destinations), x, per unit of time, namely, $\bar{v} = \frac{\Delta x}{\Delta t}$. Correspondingly, $\bar{V}$, is the average velocity for the n probes in a certain sub-stream and is defined as, $\bar{V} = \frac{1}{n}\sum_{i=1}^{n} \bar{v}_i$. Thereby, the ratio $R = \frac{\bar{V_T}}{\bar{V_R}}$ of the velocity for a testing $(\bar{V_T})$ and a reference $(\bar{V_R})$ period of time is the relative success of probes completing traceroutes. In our experiments we compute the reference

#Outages (out of 53)	Percentage of /24s also reported by Trinocular	R_outage (Average velocity ratio during outage)	Average outage duration (in hours)
23	100%	0.16	1.23
10	71-99%	0.22	1.39
4	51-70%	0.48	1.15
7	1-50%	0.31	1.13
9	0%	0.25	0.8

TABLE II: Outages reported by Disco in 2015 compared to Trinocular results.

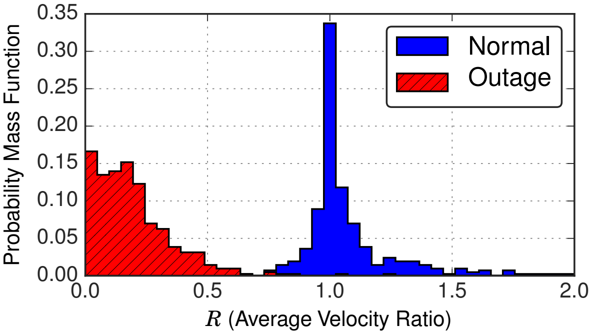

Fig. 3: Distribution of Average Velocity Ratio for normal and outage durations.

velocity $\bar{V_R}$ with traceroutes obtained one hour before and one hour after detected outages. We assume similar velocity for two periods of time representing normal operation; thus the value of R is expected to be around 1. During an outage, however, the average velocity is drastically decreased, hence, R is expected to be close to 0. This enables us to determine whether the events singled out by the detector correspond to outages.

We calculate *R* for the 443 intervals where Disco detected an outage versus normal operation for the same set of probes. Figure 3 shows that under normal conditions R is usually 1 while during an outage far fewer traceroutes complete, giving R values closer to 0. In Figure 3 we observe that all selected normal periods of times have velocity higher than the midpoint value, $R = 0.5$. Using this midpoint we determine if a reported event is considered as true positive ($R \leq 0.5$) or false positive ($R > 0.5$), and we obtain a precision of 95% for Disco.

B. Comparison to Trinocular

For a radically different view, we compare outages detected by our method with the Trinocular outage survey dataset. Disco detects 53 outages from April to December 2015, the period covered by both Trinocular and the RIPE Atlas metadata of interest. We compare the affected network prefixes revealed by Disco to the Trinocular observations for these prefixes.

First, we extract prefixes of the probes that belong to bursts of disconnect events. Then we query the Trinocular data for the same /24 prefixes to check if an outage was detected at the same time window.

We note that Disco also detects prefixes that are part of an outage that Trinocular could not probe as they are unresponsive to ICMP traffic. Out of 851 /24s (from the 53 outages in consideration) detected by Disco, only 365 (43%) were pinged by Trinocular. This shows the potential of Disco to detect outages in places where state-of-the-art active probing cannot reach. Moreover, due to use of *aggregations*, unlike Trinocular, Disco not only provides the prefixes that suffered an outage but also points out set of prefixes that are part of the same outage. Only the common 365 /24s from 53 outages could be considered for further validations.

We observe in Table II that for the /24s reported by Disco and also probed by Trinocular, both detectors agree on the down status of all the prefixes for 23 out of 53 (43.4%) outages (top row). There are some cases where Disco detected some prefixes affected by an outage but Trinocular did not. In general, about 62% (33 out of 53) of outages agree on the status for more than 70% of the prefixes. Surprisingly, there were 9 outages (16.98%) reported by Disco, with prefixes probed by Trinocular, that are not reported by Trinocular (see Table II bottom row). As explained above, we look at pre-existing traceroutes from probes to check if they indeed lost connectivity during these outages. As shown in the third column of Table II the drop of average velocity (R) during these periods of time indicates that these are indeed outages missed by Trinocular. We verified that low R values are obtained for the 9 events. In addition, the last column of Table II shows that the average outage duration for these cases is significantly lower than other cases, suggesting that Trinocular missed short-lived outages.

We also investigated 32 events reported by Trinocular but not detected by Disco. We obtained these 32 events by extracting outages for all /24s visible to both detectors from Trinocular dataset. Then we aggregated outages of several prefixes into events if they started and ended in the same 30 minute window (same aggregation metric we used for Disco). The average velocity R for these events suggests that Trinocular reports 9 false positives. As Disco has reported 47 true positives (and 6 false positives) but missed 23 events found by Trinocular the recall in 2015 (April to December) is 67%. We note that the 23 events missed were due to their non-bursty nature, a use-case that Disco is not designed for.

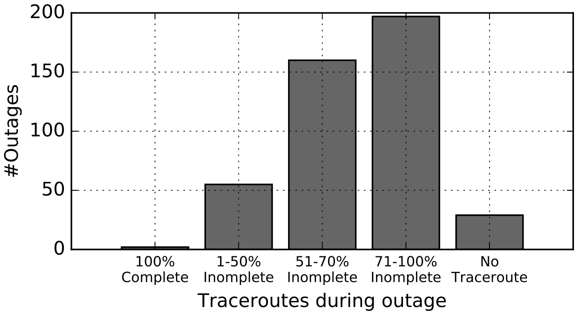

Fig. 4: Number of outages with complete, incomplete, or no traceroute.

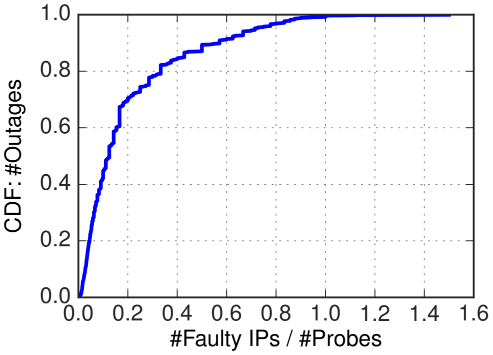

Fig. 5: CDF: Ratio of unique faulty hops to the number of probes, low ratio indicates incomplete traceroutes ending at the same hop during an outage.

VI. Outage Characterization

In this section we analyze the traceroutes initiated by probes to gain more insight into the outages we detected. Unlike previous work, with a view from outside the affected networks, we present insights from the inside, i.e., the way probes saw the outages. We use the same traceroutes as in Section V. The traceroutes are sourced by probes identified in the outages. They are buffered by probes waiting for communication to a controller to be reestablished.

The first symptom of interest is a complete lack of traceroutes reported on reconnection. We believe this is an indicator of a power outage. This is verified in the case of the power outage in Kenya (Figure 1) and for a recent power outage in Amsterdam, NL on January 18th 2017 [9]. The probes stopped operating due to the lack of power and did not perform any of the regular scheduled traceroutes.

Second, during outages traceroutes may stop at earlier hops than during normal operation. Figure 4 shows the number of outages with percentage of traceroutes during the outage that were either complete, incomplete or had no traceroutes during the outage. Except for 2 out of 443 outages, there is usually a large fraction of incomplete traceroutes during the outage or no traceroute at all. In most cases, 71-100% of the traceroutes conducted during the outage were incomplete. This is a convincing sign that probes lost complete connectivity at the detected time.

For cases with 1-50% and 51-70% incomplete traceroutes, we found that some probes kept a limited connectivity and are still able to reach local targets, such as, anycasted root servers located near the probes.

We also investigated the 2 events where all traceroutes were complete. On closer inspection we notice these events had only 2 traceroutes each conducted during the outages and these 4 traceroutes were within 3 minutes of the outage end estimated by `Disco`. As stated in Section IV-C, to estimate the end of the outage we assume that at least half of the probes involved in the burst should reconnect. In this particular case, a few probes got connected and conducted 4 successful traceroutes within 3 minutes before our estimated outage end.

Next, we characterize where incomplete traceroutes end (i.e. inside or outside probes' local ASs), and study incomplete traceroutes due to forwarding loops.

Narrowing down the location of the outage: To identify faulty hops in incomplete traceroutes, we employ the probabilistic model proposed in [14]. In a nutshell, traceroutes before the outage are used to learn the visited IP addresses at every hop and to construct a probabilistic graph of the IP paths for each destination. Given an IP address in this graph we can then estimate what would be the next visited IP address. Thus, by looking at the last hops in incomplete traceroutes we estimate the addresses that are expected on the path. This technique is, however, not able to find unresponsive addresses if the last hop was not discovered during the learning phase, for example, a new route taken during an outage. For 80% of the outages we could estimate faulty next hops for up to 60% of the traceroutes in those outages.

For a given outage, we found that the incomplete traceroutes for each destination (DNS root server or anchor) usually end at the same estimated next hop regardless of the originating probe. For each destination, if traceroutes from *n* probes were incomplete, we estimate all the next hops and compute the ratio of the number of unique faulty hops to *n*. If all probes see the same faulty hop then this ratio is low. A CDF of this ratio is shown in Figure 5. In more than 80% of the cases, this ratio is lower than 0.35, indicating traceroutes from different probes to a given destination usually failed at the same hop. This, and the fact that the identification of expected hops works most of the time, means that in most cases we are able to discover precisely beyond which IP the outage occurred. Using the faulty hops we can also distinguish between outages occurring in the probe's AS from those occurring outside of that AS. Out of all the incomplete traceroutes, 73.5% failed outside the probe's AS and 26.5% within the probe's AS.

Outages with forwarding loops: We were surprised to find numerous incomplete traceroutes caught in *forwarding loops*. These are easily identifiable as IP addresses re-appearing in the same traceroute at different hops.

In Figure 6 we show the CDF of the number of outages we detected versus percentage of traceroutes during that outage which showed a forwarding loop. We observe that for 80%

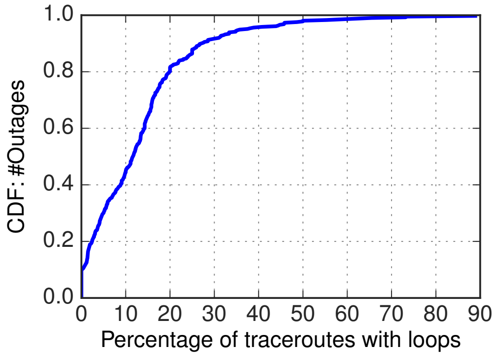

Fig. 6: Distribution of the percentage of traceroute with forwarding loops per outage.

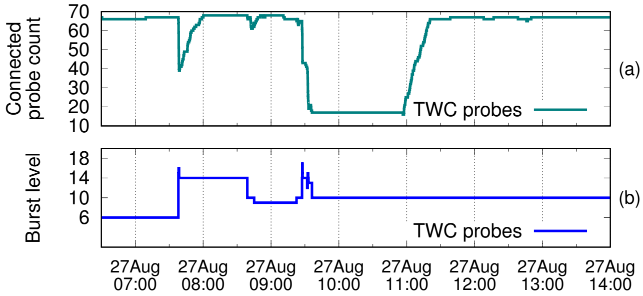

Fig. 7: TWC outage: Probe counts and corresponding burst levels.

of the outages up to 20% of the incomplete traceroutes are caused by forwarding loops.

We keep track of IP addresses involved in loops, namely, pairs of adjacent IPs in the loop. For example, in traceroute {ip1-ip2-ip3-ip2}, {ip3,ip2} characterizes the observed loop. We observe that for 80% of the outages, when a forwarding loop is seen, up to 70% of the traceroutes during that outage see the exact same forwarding loop.

The characterizations above indicate the types of outages our method can detect. ISPs often have means to detect outages in their network but they have limited visibility into what happens in neighboring networks. The ability to detect large outages that occur in another AS, locate unreachable IPs, and detect forwarding loops motivates the use of Disco for ISPs willing to better diagnose reachability issues.

VII. Case Studies

We focus on two outages in the Time Warner Cable (TWC) network and a power outage in Amsterdam. These use cases serve as examples of what network operators and researchers can learn about an outage using Disco on RIPE Atlas data stream.

A. Outages in TWC

The first outage is identified on August 27th 2014 [3]. During this outage the burst level of 17 was reached, indicating a very sudden drop in number of connected probes (Figure 7), this outage particularly stood out in the 6 years of data. It appeared in the AS-level sub-streams of AS10796, AS11351, AS11426, and AS20001, all belonging to TWC. Before the large outage of about 2 hours starting at 09:30 UTC, Disco also detected a burst of disconnection at 07:30 UTC. This outage was much shorter than the following one, but, we believe the alert at 07:30 could have been used by network operators as an early warning before the larger outage at 09:30. The outage characterization with traceroutes buffered during the outage reveals that 73% of the traceroutes that failed, suffered a forwarding loop. We also could pinpoint areas of fault by locating the common failure points of the traceroutes. Probes from Honolulu, Hawaii could reach up to Pittsburgh and probes from LA, San Diego could reach up to San Jose.

Disco also reported an outage on December 27th 2015 for AS11426 sub-stream and a geoProximate sub-stream in North Carolina. TWC announced that a router issue had been identified in this area [2], affecting users' connectivity. This example illustrates the ability of Disco to precisely identify local network connectivity issues.

B. 2017 Outage in Amsterdam

On January 17th 2017 Amsterdam suffered a large power outage. As this area has one of the highest probe densities it is another interesting case-study that shows the benefits of the geoProximate sub-streams. Figure 8, shows in red the 56 disconnected probes we detected in geoProximate sub-streams and in green all other connected probes in the area. A large proportion of reported probes is concentrated within the boundaries of the cities affected by the power outage. Interestingly 19 of the probes in that disconnect burst are outside of the city boundaries. All these probes are hosted in a single network. Traceroute data and contact with the network operators revealed that, while these probes stayed physically powered, their Internet connectivity was disrupted between two network elements in the Amsterdam area, coinciding with the Amsterdam power outage. The operators of the affected network speculate that either a network element that terminates user sessions got overloaded by having to disconnect users in the power-outage affected area, or the network between these two network elements, which is opaque to the network operator in this case, got disrupted. The fact that Disco's geoProximate sub-streams emphasized this, shows that we capture real events and interesting side-effects of outages in confined geographic areas.

VIII. Conclusion

Disco is a light-weight outage diagnostic system based on detecting bursts of TCP disconnects and generating no new measurement packets. It allows network operators to monitor over the full extent of their network, from beyond customer premise equipment up to, and including, upstream networks, regardless of whether ICMP probing is allowed in the relevant network. Using Disco we monitored the long-running connections between the RIPE infrastructure and Atlas probes. We found that 25% of the studied /24s

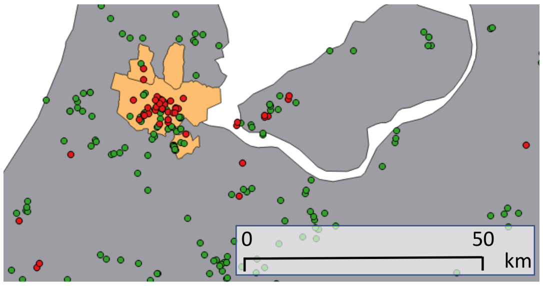

Fig. 8: Amsterdam power outage. Probes detected using geo-Proximate sub-streams (red) and connected probes (green).

are unresponsive to active probing techniques, thus proposed method contributes significantly to the current community outage detection systems.

Our work goes beyond detecting outages. Multi-resolution analysis, i.e. geographical and topological sub-streams, provides information about affected AS, country or city-level radius. This extra knowledge can be a huge asset in localizing areas of impact and potentially reduce response time.

We use existing traceroute data from Atlas probes to validate our results and to characterize detected outages. `Disco` achieves a precision of 95% and detects both outages that happen inside and outside the Atlas probe ASs. Post-mortem analysis of existing Atlas traceroutes not only helps determining common network elements where failure was concentrated but also reveals interesting characteristics such as forwarding loops. Understanding these cases helps to identify routing configurations that may go amok.

Our work also opens other interesting research questions such as studying partial connectivity during outages. Visibility into failures that occur outside the probe (customer) AS can reveal where ISPs need to focus on infrastructure development. In the future we aim to run `Disco` live and report on outages in near real-time.

References

[1] The ams-ix outage as seen with ripe atlas. https://labs.ripe.net/Members/kistel/the-ams-ix-outage-as-seen-with-ripe-atlas.

[2] Router Issue Caused Time Warner Cable Outage in the Carolinas. http://www.twcnews.com/nc/triangle-sandhills/news/2015/12/27/time-warner-cable-outages-reported-statewide.html.

[3] Time warner cable outage. https://labs.ripe.net/Members/emileaben/time-warner-cable-outage.

[4] UCSD Network Telescope. https://www.caida.org/projects/network_telescope/.

[5] USC/LANDER Project. Internet Outage Dataset, PREDICT ID: "USC-LANDER/internet_outage_adaptive_a[20-22]all". http://www.isi.edu/ant/lander.

[6] 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 *International Conference on Passive and Active Network Measurement*, pages 206–219. Springer, 2015.

[7] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207–211, 2005.

[8] BBC. Kenya nationwide blackout caused by rogue monkey. http://www.bbc.com/news/world-africa-36475667.

[9] D. P. Broadcasting. Power failure in Amsterdam, 2017. http://nos.nl/artikel/2153383-stroomstoring-regio-amsterdam-na-uren-opgelost.html.

[10] A. Dainotti, K. Benson, A. King, B. Huffaker, E. Glatz, X. Dimitropoulos, P. Richter, A. Finamore, and A. Snoeren. Lost in Space: Improving Inference of IPv4 Address Space Utilization. *IEEE Journal on Selected Areas in Communications (JSAC)*, 34(6):1862–1876, Jun 2016.

[11] A. Dainotti, C. Squarcella, E. Aben, K. C. Claffy, M. Chiesa, M. Russo, and A. Pescapé. Analysis of country-wide internet outages caused by censorship. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, IMC '11, pages 1–18, New York, NY, USA, 2011. ACM.

[12] M. Djatmiko, D. Schatzmann, X. Dimitropoulos, A. Friedman, and R. Boreli. Federated flow-based approach for privacy preserving connectivity tracking. In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pages 429–440. ACM, 2013.

[13] M. Djatmiko, D. Schatzmann, A. Friedman, X. Dimitropoulos, and R. Boreli. Privacy preserving distributed network outage monitoring. In *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, pages 69–70. IEEE, 2013.

[14] R. Fontugne, E. Aben, C. Pelsser, and R. Bush. Pinpointing delay and forwarding anomalies using large-scale traceroute measurements. *CoRR*, abs/1605.04784, 2016.

[15] R. Fontugne, K. Cho, Y. Won, and K. Fukuda. Disasters seen through flickr cameras. In *Proceedings of the Special Workshop on Internet and Disasters*, page 5. ACM, 2011.

[16] B. Huffaker, M. Fomenkov, and k. claffy. Geocompare: a comparison of public and commercial geolocation databases - Technical Report . Technical report, Cooperative Association for Internet Data Analysis (CAIDA), May 2011.

[17] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson. Studying black holes in the internet with hubble. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'08, pages 247–262, Berkeley, CA, USA, 2008. USENIX Association.

[18] J. Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[19] J. Kleinberg. Temporal dynamics of on-line information streams. *Data stream management: Processing high-speed data streams*, 2006.

[20] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins. On the bursty evolution of blogspace. *World Wide Web*, 8(2):159–178, 2005.

[21] L. Quan, J. Heidemann, and Y. Pradkin. Trinocular: Understanding internet reliability through adaptive probing. *SIGCOMM Comput. Commun. Rev.*, 43(4):255–266, Aug. 2013.

[22] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[23] RIPE NCC. RIPE Atlas. https://atlas.ripe.net.

[24] RIPE NCC. RIPE Atlas Built-in Measurements. https://atlas.ripe.net/docs/built-in/.

[25] RIPE NCC. RIPE Atlas Result Streams. https://atlas.ripe.net/docs/result-streaming/.

[26] D. Schatzmann, S. Leinen, J. Kögel, and W. Mühlbauer. *FACT: Flow-Based Approach for Connectivity Tracking*, pages 214–223. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[27] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang. Planetseer: Internet path failure monitoring and characterization in wide-area services. In *Proceedings of the 6th Conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, OSDI'04, pages 12–12, Berkeley, CA, USA, 2004. USENIX Association.