ABSENCE: Usage-based Failure Detection in Mobile Networks

Binh Nguyen[†], Zihui Ge[‡], Jacobus Van der Merwe[†], He Yan[‡], and Jennifer Yates[‡]

[†]School of Computing, University of Utah [‡]AT&T Labs - Research

{binh,kobus}@cs.utah.edu {gezihui,yanhe,jyates}@research.att.com

ABSTRACT

We present our proposed ABSENCE system which detects *service disruptions* in mobile networks using aggregated customer usage data. ABSENCE monitors aggregated customer usage to detect when aggregated usage is lower than expected in a given geographic region (e.g., zip code), across a given customer device type, or for a given service. Such a drop in expected usage is interpreted as a sign of a potential service disruption being experienced in that region / device type / service. ABSENCE effectively deals with users' mobility and scales to detect failures in various mobile services (e.g., voice, data, SMS, MMS, etc). We perform a systematic evaluation of our proposed approach by introducing synthetic failures in measurements obtained from a US operator. We also compare our results with ground truth (real service disruptions) obtained from the mobile operator.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network monitoring, Network management

Keywords

usage-based failure detection; mobile networks; large scale; operational networks

1. INTRODUCTION

The proliferation of sophisticated mobile devices like smart phones, tablets and wearable devices [9] have made them an integral part of today's society. The growth in both the number of mobile devices, the data usage of each device and the types of mobiles devices of course implies an increased reliance and dependence on mobile networks. To address this demand, mobile operators are continuously investing in new mobile networks and technologies. In recognition of the importance of the underlying network, mobile operators are building redundancy into nearly all components of their infrastructure and developing sophisticated systems to monitor the health of their networks and to rapidly respond to any customer impacting events [15].

Despite these efforts, the inherent complexity of mobile networks and their environments (customer devices, applications) may result in service disruptions that go undetected by monitoring the network elements. Hence modern mobile operators adopt the strategy of deploying *service* monitoring, in addition to the *network* monitoring, on the customer service experience. Service monitoring is designed to continuously monitor the end-to-end experience that customers receive from their network-based services. This contrasts with network monitoring, in which the status of individual network elements and links are monitored for failures and impairments (e.g, link losses). Service monitoring is vital as a second line of defense – capturing network, customer device or application issues as well as interaction issues among them which may not be detected by the network/applications/devices themselves. Service monitoring is also crucial to quantifying the service impact of known network problems for prioritizing issue resolution.

Somewhat counter-intuitively, network elements that support the service functions are not always able to alarm on conditions which are in fact service impacting. This may be the result of, for example, software bugs in the network elements' firmware, or in the EMS (element management system) for the network elements, or due to configuration errors. It is possible that, even though all network metrics indicate a healthy network, customers might be experiencing degraded service or a complete service disruption. For example, the deployment of a new service feature or a software upgrade to address a bug, might trigger an unintended side effect (or indeed a new bug) that the monitoring system is not equipped to detect. We define such service disruptions that are not captured by network monitoring as *silent failures*. Furthermore, it is difficult to infer service quality perceived by customers using the status of the network. There is a complicated relationship between the status of the network and the service quality the users experience. For example, because of redundancy mechanisms within the network, a particular network failure does not necessarily imply customer impact. For example, users associated with a failed cell tower could be picked up by neighboring towers as long as they are in the coverage range of the neighboring towers and the neighbors still have enough resources to handle the users' traffic [27].

However, monitoring service performance across a mobile network is extremely challenging. Traditional active monitoring approaches – techniques which send test traffic across the network – simply don't scale, courtesy of the very large number of cell towers (end points) that need to be monitored and the diverse set of services supported by mobile networks. One could alternatively naively imagine looking for service disruptions by looking for drops

MobiCom'15, September 7–11, 2015, Paris, France.
© 2015 ACM. ISBN 978-1-4503-3543-0/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2789168.2790127.

in traffic volumes on network elements. However, the inherently dynamic nature of a mobile network environment makes it difficult to infer service impact by monitoring individual network elements, e.g., routers or base stations, to distinguish between changes in traffic volume that are simply the result of the normal operation of the network, and changes that are the result of anomalous network behavior.

In this paper we present our work on the ABSENCE system to address the detection of service disruptions in mobile networks in a proactive manner. Our key insight is that service disruptions often impact the traffic consumed by customers, which very likely reflects in customers' usage. This seemingly obvious observation, combined with a suitable mechanism to monitor customer usage, allows ABSENCE to rely on customer usage data to detect the possible presence of service disruptions. Specifically, ABSENCE uses aggregated (e.g., zip code level and handset manufacturer/model level) usage data for different mobile services (e.g., voice call, data, and short message) calculated from anonymized call detail records (CDRs). ABSENCE uses the historical aggregated usage data to predict the expected customer usage for different mobile services at appropriate aggregations of customers, and compares this with real time customer usage data. A deviation from the predicted customer usage is highly likely an indication of a service disruption.

We make the following contributions:

- We present the design of ABSENCE, a novel service disruption detection system for mobile networks that infers service disruptions by monitoring aggregate customer usage. Our design is informed by a data driven exploration of the problem domain using data from an operational mobile network.

- We present a scalable Hadoop-based implementation of our approach which is capable of performing service disruption detection by processing huge volumes of anonymized CDR data (e.g., hundreds of millions of records every hour for mobile data service) in a streaming manner, for all the mobile services associated with an operational mobile network.

- Using data from the same operational mobile network, we perform a systematic data-driven evaluation of our approach by introducing a comprehensive synthetic set of both network and mobile device failure scenarios. Our results show that: (i) Our variable-scale temporal aggregation improves detection by an order of magnitude over fixed interval aggregation. (ii) We achieve overall detection rates of 88%, while we achieve 98% or better detection rates for service disruption that have over 10% usage impact within the corresponding aggregation (e.g., zip code and handset device model).

- We compare our results with ground truth from actual service disruption events and present a number of case studies showing the effectiveness of our approach. For a set of confirmed service disruptions, ABSENCE achieves 100% detection rate.

2. MOTIVATION

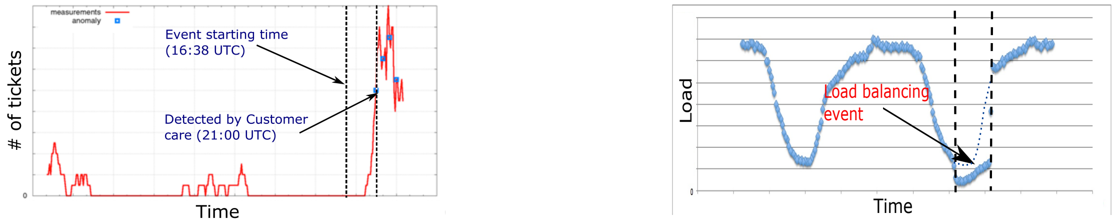
In this paper we define a *service disruption* to be a scenario in which customers become unable to utilize the offered service(s) that they would normally utilize. A service disruption can be either due to the network/application fails to complete customers' service requests, or because customers give up making service attempts due to unacceptable service performance. Service disruptions are typically the result of a network, device or application outage or severe performance degradation. The vast majority of service issues are rapidly detected via the network and/or application. However, there are a small number of issues – typically those resulting from complex software bugs – that may remain undetected by the network and/or application.

Given the challenges of scaling active service monitoring techniques, one could imagine instead simply relying on customers to inform a mobile service provider of service disruptions as is performed in other industries, such as the power industry. However, given that individual customer concerns may relate to a large number of underlying causes – individual customer device issues, customer user error or broader service disruptions – identifying a significant service disruption would typically require detecting a pattern in the customer feedback across a number of different customers. This is inherently slow, and thus a highly undesirable approach to detecting service issues. Figure 1a shows an example of customer ticket volume resulting from a service disruption. The figure shows a time series of the number of tickets in the customer care system. The actual event occurred around 16:38 UTC, but was only evident via an increase in customer ticket volumes at 21:00 UTC (i.e., 4.5 hours later), when the number of customer calls rapidly increased. Relying on customer complaints to detect such failures is thus clearly undesirable – customers are simply too slow at calling in for this to be a timely approach for detecting service issues.

Given the relative maturity of network management and operation practices [15], and significant research efforts associated with failure detection [13, 4], network and application interaction [22, 14, 19, 10], mobile network performance [20, 21, 25, 12, 6] and service monitoring [26, 24, 5], the obvious question is: *Why are these service disruptions so difficult to detect?*

Active end to end service monitoring is used extensively across mobile and wireline networks to test service integrity and performance. Active monitoring uses probes placed strategically across the network to send test traffic. However, the major challenge with active monitoring is scale – ideally probes must be deployed so that every combination of service path, customer device type and application is actively tested on an ongoing basis. But this is clearly an unrealistic expectation. There are simply too many different types of customer devices in the market place, a multitude of different applications and a huge geographic environment to probe. In a wireless environment, service performance can vary considerably across a very local region even for customers connected to a common cell site. Thus, even placing a dedicated probe associated with each individual cell site does not provide a comprehensive view of service experience across the entire region associated with that cell site. Thus, active monitoring in a mobile network provides only a sampling of service experience and, depending on the extent of the deployments, will likely not be able to detect all service impairments.

One may alternatively use passive monitoring of customer traffic to detect service impairments. Such monitoring can be performed on traffic aggregates, and thus does not need visibility into individual customer experience. However, service outage detection cannot be achieved by observing customer traffic – by its very nature customer traffic is expected to disappear during an outage. Thus, detecting a service outage using passive monitoring entails looking for an *absence* of expected traffic. Thus, one could natively assume that we could look for drops in load on network elements to identify service outages. However, where in the network to look for such reductions in carried load is an intriguing challenge – traffic is regularly shifting around a mobile network, typically without any service impact. For example, activities such as load balancing or planned maintenance events could cause load changes on network devices, yet has no impact on customers' service experience. Figure 1b shows a load change on an Serving Gateway (SGW) node in an SGW pool during



(a) Customer care tickets indicating a service disruption

(b) Load reduction caused by load balancing

Figure 1: Network events

a load balancing event. Despite the load change on the individual SGW, there was no impact on users. Simply looking at the load of this SGW alone is insufficient to determine customer impact.

While detailed network performance metrics have been defined for mobile networks by 3GPP [3], and are being implemented by equipment vendors, these key performance indicators (KPIs) are not always sufficient to detect user impact either. For example, 3GPP defines *accessibility* as a KPI to measure the probability that a user will be provided with radio access network (RAN) resources (technically with a radio access bearer) on request. This metric can clearly provide insight concerning resource shortages in the radio access network. However, users might still be impacted, even when the RAN accessibility KPI is good, because users that lost radio coverage are not even accounted for in the KPI calculation or when the root cause of the service problems are beyond the RAN (e.g., congestion on a core network element). In other words, service disruption occurs when accessibility is bad, but accessibility being good does not necessarily imply that service is good.

Finally, in order to deliver the end-to-end service successfully, handset device, mobility network and application need to work together seamlessly. Thus the root cause of a service disruption could be well beyond the mobility network. For example, a firmware upgrade on a certain customer device model could result in incompatibility between equipment in the RAN (e.g., a radio network controller (RNC)), and the mobile devices that performed the firmware upgrade, thus resulting in a large number of devices not being able to access the network. Similarly, an update in the delivery protocol of a video streaming application could cause interoperability issues between application and network, which further leads to service disruptions. These types of service disruption are challenging for mobility network operators since there is little evidence of it on the network side.

In this paper, we argue that users' usage, or lack thereof, is a reliable indicator of service outages and severe performance degradations in a mobile network. By monitoring and analyzing users' usage, we are able to detect service disruptions that could be challenging for other event detection mechanisms.

3. APPROACH

With ABSENCE we propose to use historical customer usage data to predict expected usage that should be generated from customers under normal conditions. Any deviation from expected customer usage indicates an anomaly, which might be indicative of a service disruption. While this basic approach conceptually seems to make sense, it is not obvious that the method would be feasible in practice. For example: would customer usage predictions based on usage data be sufficiently accurate to allow anomaly detection? Given the number of mobile devices, the variety of services offered on mobile networks and the complexity of the mobile network infrastructure, is there a level of customer usage aggregation that would provide fidelity of detection fine-grained enough to detect silent failures at a granularity that is practically useful? Given that users in a mobile network are by design using the network while moving around, how should we deal with mobility?

In this section, we present our exploration of these questions using customer usage data from a large mobile network provider. Before describing our exploration we briefly describe the nature of Call Detail Records (CDRs) which, when aggregated, constitute the customer usage data used in ABSENCE.

3.1 Customer usage data

ABSENCE aggregates metrics from Call Detail Records (CDRs) to measure customer usage at different locations and application levels. These aggregates are calculated within ABSENCE using individual, anonymized CDRs. Note that these individual CDRs are only used internally to the system during the aggregation process, and no customer specific data (anonymized or otherwise) is ever exposed to a user of the ABSENCE system. Call Detail Records contain meta data about executed transactions across the mobile network (i.e., phone call, data session, access to voice mail etc.). Each record captures information that is needed for charging and debugging such as a time-stamp of the activity, device specific information (e.g., the international mobile station equipment identity (IMEI)), network related information concerning the activity (e.g., the sector(s) of a cell tower that the device is connected to), the duration of the activity (for voice services) or the volume of data the device downloads/uploads (for data services). Of critical importance for our approach, CDRs are generated in near real time: for Voice service a CDR record is generated right after a call finishes, for Data service a CDR record is generated whenever a PDP context is created and a new CDR record is created every hour if the data connection spans multiple hours. This allows CDRs to be used as a timely indicator of customer activity (or inactivity). Note that CDRs do not contain actual customers' short message, voice call or data content and ABSENCE only uses anonymized CDRs.

3.2 Usage prediction and aggregation size

The primary challenge in calculating aggregate customer usage information is to determine what level of aggregation is most effective to address the problem at hand. In considering the question of accuracy of predicting traffic volumes versus fidelity of anomaly detection, there exists an intuitive tradeoff: At one extreme one might attempt to use the usage data of each individual. While the network usage pattern of an individual user might show fairly predictable patterns, at this granularity a deviation from an expected pattern is clearly not a reliable indication of a service impairment. The user in question might simply have a change in their normal behavior, e.g., going on vacation. At the other extreme, the usage aggregated across all customers in the US is highly predictable. However, at this aggregation level, a service disruption that only impacts a relatively small number of users, e.g., those associated with a particular cell-tower, would not be visible at such a coarse grained data aggregation level. The challenge therefore is to find an aggregation level of usage data that is small enough that it can provide high fidelity of detection, but at the same time large enough to render stable usage patterns to allow for accurate prediction.

For example, Figure 2a and 2b show the amount of voice calls respectively made by a group of 70 and 3020 randomly chosen devices over the course of three weeks. The amount of usage on the smaller group is significantly less that in the larger group and the larger group also shows more stable day-to-day usage pattern between the different weeks.

To understand how the amount of usage affects the tradeoff between the stability of the usage patterns versus the fidelity of detection regardless of device specific, we conduct experiments to predict future usage based on historical usage with different amounts of aggregated data.

Experiment description: We selected a uniform random sampling of users to form different groups (with size ranging from 20 to 150,000 users) for Voice and LTE data service. We assumed that the aggregated service usage follows a weekly seasonal model and we used 16 weeks of data for our training. We constructed a weekly seasonal usage pattern using the additive decomposition technique described in Section 4. The seasonal pattern is the predicted usage for the future usage and the "noise" is the absolute distance between the usage and its seasonal data point. We then used the above seasonal usage pattern to predict another week of usage.

Metric: To quantify the prediction accuracy, we use the normalized noise ratio as the metric. We formally define the noise ratio in Section 4. In short, the noise ratio is the "noise" between the testing data and the training data normalized by the training data. Intuitively, if the usage of an hour deviates too much from the seasonal pattern, the noise of that hour is high and therefore results in a higher normalized noise ratio. We sampled the noise ratio at two regions of a time series (i.e., during the peak usage period 17:00-23:00 UTC and during the low usage period 03:00-11:00 UTC), and plot the noise ratio as a function of usage.

Results: Figure 2c shows the noise ratio as a function of the usage for LTE and voice. Overall, for a sufficient aggregation (i.e., above 1,000 of usage), the usage is quite predictable (i.e., the noise is about 10%). Moreover, the noise ratio is high for a small usage and reduces when the amount of usage increases. This matches the intuition that the usage of an individual user is less predictable than the aggregated usage of a group of users. The figure also suggests that after a certain amount of usage data, the predictability of an aggregation does not increase significantly. This suggests that the size of an aggregation should not be too large for both good predictability and high sensitivity, e.g., if we monitor usage of an entire city as an aggregation, a failure that impact only a single ZIP code area might not cause a significant enough drop on the total usage for a system to detect.

3.3 Practical user aggregation

In this section we consider the question of how groups of users can be selected in practice. While network and service failures can be highly diverse in their impact scope, the mobile network design and operational practice would inherently cause the service failures to be localized to geographically close-by regions and/or user devices with some common hardware or software. For example, rolling out a software upgrade on Radio Network Controllers (RNCs) would typically take place in a few geographical regions, and the upgrade may introduce an unexpected compatibility problem with certain phone models that was not captured by the RNC equipment vendor during lab testing. As another example, a software bug in a packet data network gateway (PGW) may cause the service that the PGW supports (e.g,. visual voice mail) becomes unusable, and all service requests originated from certain geographical regions that are routed toward this PGW are affected. Hence, grouping users geographically and by device hardware and software and tracking usage by different service features would have the best chance of capturing service failures.

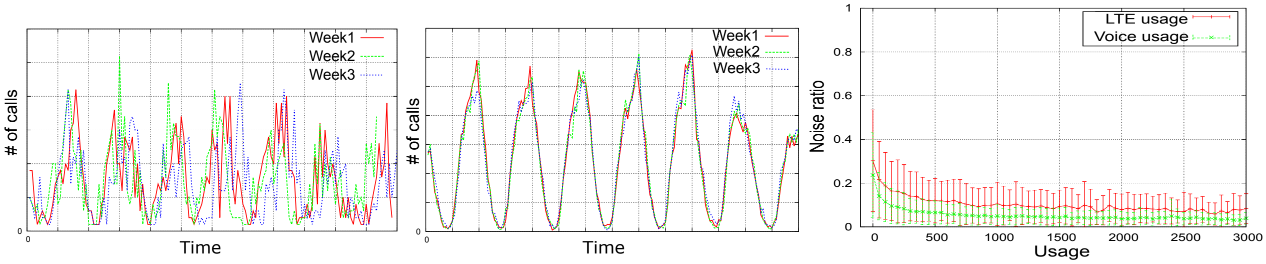
Geographical hierarchy: In this grouping method, we utilize the geographical hierarchy in the ZIP-code system to group users. We use the ZIP-code hierarchy for three reasons: (i) the ZIP-code system was designed for efficient postal delivery and therefore each ZIP-code naturally covers a sufficient and relatively equal amount of users, (ii) the ZIP-code hierarchy is geographically driven and the structure of the ZIP-code has geographical meanings, (iii) by utilizing the ZIP-code hierarchy, the system can quickly scale up and down the size of aggregation based on the structure of the ZIP-code. I.e., groups of states, states, large cities etc. Moreover, a ZIP code area is relatively large enough for sufficient usage prediction and small enough to obtain good sensitivity A ZIP code area also often belongs to either an urban or a rural area and therefore users in a same ZIP code often have the same usage pattern. The detailed ZIP-code hierarchy is presented in Figure 3a.

Device type hierarchy: Under each geo-group (i.e, a node in Figure 3a), we divide devices into smaller groups based on operating systems (i.e, Android, IOS, Windows, BlackBerry OS), device make (i.e, Samsung, Apple, Nokia, etc), and device type (i.e, Samsung Galaxy S5, iPhone 4, Nokia Lumia 512, etc). This way the system can monitor not only geographical aggregations (e.g., Salt Lake City) but also specific device types in the area (e.g., all Samsung Galaxy S4 devices in Salt Lake City). As shown in Figure 3b, this device hierarchy can be applied at different levels in the geographical hierarchy.

3.4 Temporal usage aggregation

Recall that in Section 3.2 we found that in order to get good predictability the aggregations being monitored should have a large enough usage. (Figures 2c.) Daily network usage follows a well known diurnal pattern with well established busy and quiet times. During the network quiet time (i.e., after midnight), hourly usage is typically small and might not be sufficient for a good predictability. Figure 3c shows the CDF of hourly voice usage during low usage period (i.e., 03:00 UTC to 11:00 UTC) and peak usage period (i.e., 17:00 UTC to 23:00 UTC) of all ZIP codes. Almost 95% of the hourly usage measurements during low usage period are smaller than 500 which reduces the accuracy of prediction over hourly aggregations. I.e., at the ZIP code level, simply grouping usage into hourly bins results in insufficient usage to obtain a good prediction. In contrast, 48% of the hourly usage measurements during peak period are smaller than 500. Moreover, usage can also be low if, when using the geographical or device aggregation, the chosen aggregation level only has a small number of users (e.g., a ZIP code in a rural area or a ZIP code with a small number of subscribers).

This suggests the utility of grouping multiple hours during low usage period or of small spatial aggregations into a single temporal aggregation. Note that using longer time periods over which to do the usage aggregation would present higher accuracy at the cost of increasing the potential detection time, i.e., when the usage is low this technique increases the likelihood of detecting a failure after several hours while the conventional technique cannot detect it. Based on this observation we employ a *variable scale aggregation* strategy in Section 4 to improve the accuracy of prediction in ABSENCE.

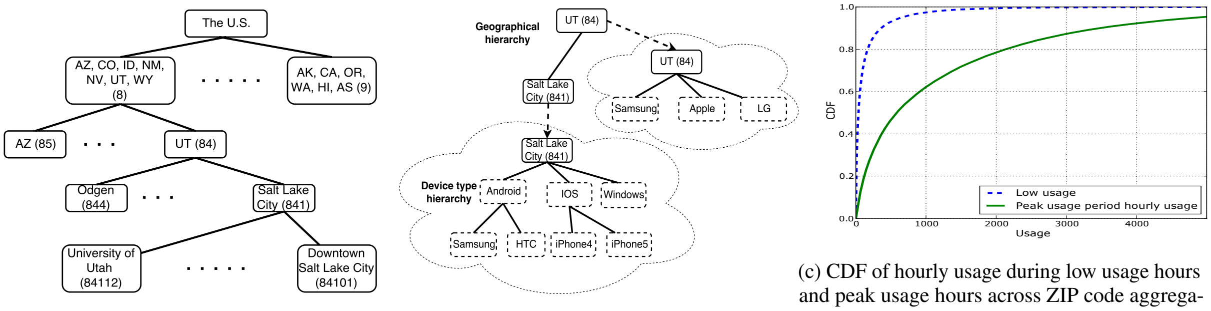


(a) Usage of group of 70 devices

(b) Usage of group of 3020 devices

(c) Noise ratio of LTE and voice usage

Figure 2: Usage prediction and noise ratio



(a) Geographical hierarchy

(b) Device hierarchy

(c) CDF of hourly usage during low usage hours and peak usage hours across ZIP code aggregations

Figure 3: Hierarchies and hourly usage

3.5 Usage aggregation

ABSENCE uses aggregate service usages to detect service disruptions. In order to generate these aggregates, ABSENCE groups individual anonymized CDRs from a set of similar users. In ABSENCE, users are considered similar to each other if their mobile devices are from the same manufacturer/model or from the same zip-code. Aggregation usage from users with the same mobile device manufacturer/model is straight-forward. Due to space constraints, in the section, we only focus on how to group users based on the zip-code that they are in.

The inherent challenge stems from the mobility nature of mobile network users. Thus the set of users in a particular zip-code area, is changing over time. For example, the number of mobile users in a zip-code covering a business or educational campus might vary by thousands, or even tens of thousands, over the course of a day as workers/students arrive for the work day and leave again at the end of the day. In this section we explore mechanisms to deal with this inherent variability in ABSENCE.

Our approach hinges on the observation that for our purposes in ABSENCE, the exact location and mobility patterns of specific users at a particular point in time are not relevant. Rather, we are interested in knowing these properties with sufficient accuracy for a statistically meaningful aggregate of users. Moreover, approximated users' location could be used to localize geographical location of a failure to benefit root cause analysis. Thus, to calculate the zip-code level usage aggregates we simplify the internal operation of our ABSENCE system by simply aggregating over those customers that are typically in the given zip code for that time of day. Thus, within ABSENCE, for each (anonymous) user, we derive a *user zip-code profile* which approximates the user's mobility pattern at zip-code level over time. This user zip-coed profile is derived within ABSENCE using the anonymized CDRs. After calculating an individual user profile, the usage of an user is counted toward *his/her profiled zip-code* regardless of the current zip-code of the user. For example, during summer some students leave their campus for internship, those students' usage will be counted toward their approximated zip-code, i.e., their campus.

Zip-code level profiling: We evaluated a number of strategies to derive the user profiles. We assumed users' mobility pattern follows a daily pattern during weekdays and weekends (i.e., weekends are treated differently). We explored 2 parameters used to approximate users' zip code profile: how many zip-codes in a day a user has and the length of training data used for the approximation. Note that maintaining multiple zip-codes for a user results in more resources required to store the historical data and extract usage in real time. For example, if the user zip-code profile maintains 48 zip-codes for each user (i.e., 24 hours for weekdays and 24 hours for weekends) then there are 48 historical usage measurements for that user and each of them may be grouped differently (recall that ABSENCE groups usage based on user zip-code profile).

Due to space constraints we only describe the *home/work* approach which provided acceptable complexity/accuracy tradeoff and is what we use within ABSENCE. Our experience is that the fine-grained hourly user zip-code estimation does not improve the accuracy yet requires significant more computation. With the home-/work profile approach we make the simplifying assumption that user mobility can be approximated as follows: Depending on the time of day and day of week, a user is typically either at home or at work. Specifically, during week days and working hours, i.e., 9 a.m. to 7 p.m., the user is assumed to be at work. While during weekends and the remaining week day hours, the user is assumed to be at home. Given this assumption, what is required to derive a home/work profile is to determine the user's home and work base station. We make use of historical CDR data over a relatively long period of time, i.e., a couple of months, and simply use the most frequent base stations for the appropriate time (i.e., work or home) to determine the respective base stations for home and work hours. For example, if a user uses a base station most frequently during 9 a.m. to 7 p.m in a month period, his/her work hour zip-code profile will be that base station's zip-code. The home/work profile is clearly quite scalable requiring only two zip-codes (and associated historical information) to be maintained for each user.

Experiment: We evaluated the accuracy of the home/work zip-code level profile. We first derived the users' zip-code profiles as described above using historical CDRs. We varied the amount of the

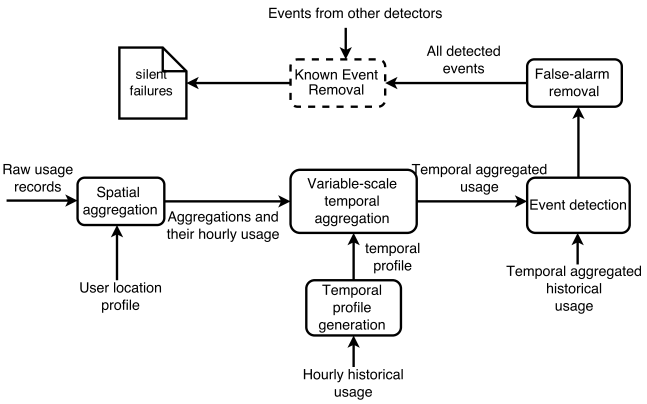

Events from other detectors
All detected events
silent failures
Known Event Removal
False-alarm removal
Raw usage records
Spatial aggregation
Aggregations and their hourly usage
Variable-scale temporal aggregation
Temporal aggregated usage
Event detection
temporal profile
User location profile
Temporal profile generation
Temporal aggregated historical usage
Hourly historical usage

Figure 4: Data processing pipeline of ABSENCE

training data used to evaluate its impact on our approximation by 1,2,4, and 8 weeks. We then used the derived profiles to predict users' zip-code using the same amount of future CDRs. We measured the percentage of users' zip-codes that were correctly predicted (i.e., the hit rate - whether he/she makes calls in the estimated home/work zip-code) using the user profiles.

Results: Table 1 shows the results of this evaluation. As shown the duration of the historical (training) data does not significantly improve the accuracy of the approximation. Given the computational cost to consider this suggests that the home/work zip-code profile, with monthly updating, will be the most suitable for ABSENCE.

	1-week training	**2-week training**	**4-week training**	**8-week training**
Home/work	57.03%	57.92%	*58.91%*	56.05%

Table 1: Hit rate of zip-code approximations

4. SYSTEM OVERVIEW

In this section we describe ABSENCE and how the data is processed. The logic blocks of the system are shown in Figure 4. From left to the right, "raw usage data" (anonymized call detail records) go through a pipeline, which consists of multiple stages: spatial aggregation, variable-scale temporal aggregation, event detection and false-alarm removal before the detected events are shown to the operators. While we show this step for completeness in Figure 4, our current focus is on developing an effective usage-based detection system and we therefore do not consider this aspect. In Section 8 we do, however, validate ABSENCE against real service disruptions.

Spatial aggregation: We explored the need for spatial aggregation to ensure the accuracy of estimation in Section 3. As we described in Section 3.3, ABSENCE groups users based on their profiled zip-codes (Section 3.5): users that are geographically close are grouped together using their zip-code profile. Users in the same area are often being served by the same set of network elements and therefore likely to be impacted as a group. Moreover, events that impact a group of geolocated users are more likely to be actionable to network operators.

Temporal profile generation: After grouping the users, ABSENCE needs to extract the usage data of the group over time in order to detect abnormal usage for the group. As mentioned in Section 3.4, simply grouping usage into hourly bins is not optimal for usage predictability and anomaly detection. Instead, if the hourly usage is smaller than a predefined threshold (e.g., after midnight or the spatial aggregation is small), ABSENCE groups multiple hours of usage into a single bin in order to satisfy the usage threshold.

To realize this, ABSENCE needs a *temporal profile* for each spatial aggregation to do the grouping. ABSENCE assumes a weekly seasonality for the aggregations, i.e., the usage of an aggregation repeats every week. To generate the temporal profile, ABSENCE first uses hourly historical data to find a regressed weekly time series such that every data point in the regressed weekly time series is the *median* of the historical data points. Note that the historical time series and the regressed weekly time series consist of hourly usages. Having calculated the regressed weekly time series, ABSENCE then runs a greedy algorithm (Algorithm 1) that groups consecutive hours together until the total usage is larger than a predefined threshold (i.e., K in algorithm 1) and repeats this until all the hours in the weekly time series are grouped. If the last temporal bin of a week appears to be too small then it will be combined with the first hours of the following week until the threshold K is satisfied. The output of the algorithm is a *temporal profile* which has each temporal bin is at least the predefined amount of usage. Note that ABSENCE needs to run this training process only once every several months given that the *temporal profile* of the aggregation is usually stable.

Variable-scale temporal aggregation: After obtaining the *temporal profile* for each spatial aggregation, ABSENCE uses the profile and the hourly time series from the spatial aggregation to create a variable-scale time series. The output of the *variable-scale temporal aggregation* is a time series which has multiple temporal granularities and each data point satisfies a predefined usage threshold. This time series is used for *event detection*. We compare this variable-scale approach with the plain hourly temporal aggregation in Section 7.1. Note that ABSENCE currently aggregates usage data on an hourly basis at the finest granularity.[1]

Algorithm 1 Generate temporal profile

Input: Weekly regressed usage time series $T = (t_0, t_1, ..., t_{167})$ for 168 hours in a week, threshold K.
Output: Temporal profile j and $P = (p_0, p_1, ..., p_j)$ as the starting hour of j continuous segments of each profile bin

```
Initialize: $i \leftarrow 0$, $accumulate\_usage \leftarrow 0$, $j \leftarrow 0$, $tag \leftarrow 0$
while $i < 167$ do
    $accumulate\_usage \leftarrow accumulate\_usage + t_i$
    if $\sum(accumulate\_usage) \geq K$ then
        $p_j \leftarrow tag$, $accumulate\_usage = 0$, $j \leftarrow j+1$, $tag \leftarrow i$
    end if
    $i \leftarrow i+1$
end while
if $j == 0$ then
    return 0, ()
else
    if $accumlate\_usage == 0$ then
        return j, P
    else
        {remainder wrapping around to the beginning of the week}
        $p_0 \leftarrow tag$
        return j, P
    end if
end if
```

Event detection: After generating the *variable-scale* usage time series of a group, ABSENCE appends the usage with the corresponding *variable-scale* historical usage and feeds the entire time series into a time series decomposition and event detection module that analyzes the time series and outputs abnormal events. Due to the large number of time series that needs to be processed, ABSENCE adopts the additive time series decomposition approach, which is a light-weight time series analysis algorithm and has been found very effective in modeling economic data and recently in network

[1] While an hour might seem long from the perspective of detecting a network outage, it does represent a reasonable tradeoff in the context of network operations scale.

traffic as well [23]. At a high level, the time series decomposition technique de-constructs a given time series into the secular trend component ($\{T_t\}$), the seasonal component ($\{S_t\}$), and the noise component ($\{N_t\}$) [8]. In the additive model, the original time series ($\{V_t\}$) is the summation of these three components.

Figure 5 (b), (c), (d) show the corresponding components for the time series in Figure 5 (a). The general idea of time series decomposition is very simple – with a specified seasonality window W, secular trend can be obtained through smoothing over long term (multiples of W), i.e., by *centered moving average*:

$$T_t = \sum_{i=-W}^{W-1} V_{t+i}/2W$$

Note that to be able to decompose the *variable-scale* time series, the seasonality window W here is set to the number of temporal aggregations of the *temporal profile* generated, i.e., j in Algorithm 1, and W varies for different spatial aggregations.

The seasonal trend can be obtained by averaging the phase value (after removing secular trend) across seasons, i.e., by *seasonal moving average*:

$$S_t = \sum_{i=0}^{K} V_{t-iW} - T_{t-iW}$$

where K is the number of seasonal windows contained in the historical data. And the remainder becomes the noise component:

$$N_t = V_t - T_t - S_t$$

Note that time series decomposition can be applied to analyzing both long range historical data and in a moving window fashion for the recent data (as new data is appended to the time series).

In our approach, we further model the noise components, N_t, at different phases as zero-mean Gaussian variables with different variance, $\sigma^2_{t|W}$, where the phase $t|W$ represents t mod W. We tag the corresponding time series value, V_t, as anomalous (critical value 1.96 at 95% confidence interval) if

$$|N_t/\sigma_{t|W}| > 1.96 \qquad (1)$$

This is consistent with classic anomaly detection techniques. We also apply an iterative process such that we remove the anomalous points in the previous iteration from the trends and noise variance computation, which makes our approach robust to bad data/known anomalies.

An example of how the event detection works is shown in Figure 5. The two dips (green dots) in Figure 5 (a) correspond to the two dips in the noise component in Figure 5 (d) (red solid line) and those two dips are smaller than the lower 95% confidence interval of the noise component at the points (blue dashed line). This results in two detected anomalies in the time series.

5. IMPLEMENTATION

Processing the anonymized CDRs is computational intensive (e.g., hundreds of millions of records every hour for data service and tens of millions of records every hour for voice service) and normal serial processing methodologies will not be scalable. Since CDRs can be processed independently, we use a Hadoop Map-Reduce cluster to process the data in parallel to speed up the process.

Running environment: As shown in Figure 6, Absence consists of four components: historical usage retrieval, hourly usage retrieval, time series processing and user location profile retrieval. Absence runs on two environments: usage retrieval is done on a Hadoop cluster and time series processing is done "locally". The

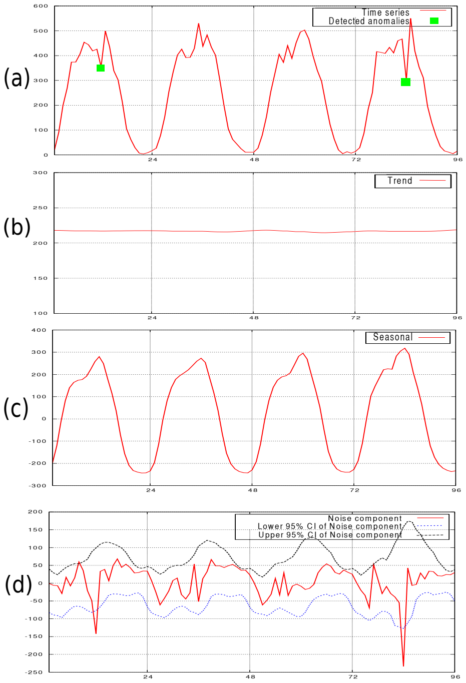


Figure 5: Trend, seasonal, and noise components

Hadoop cluster hosting Absence consists of 100 nodes each with 32 cores CPU and 128GB RAM and runs on a HDFS file system. The local environment is a single node in the Hadoop cluster.

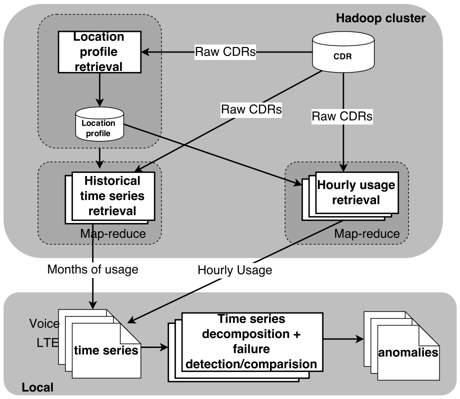


Figure 6: Absence components

Historical/hourly usage retrieval component: We use Apache Pig (pig.apache.org), a platform that offers a high-level language for expressing map-reduce programs, for the usage retrieval components. The hourly usage retrieval component wakes up every hour to extract the usage of the last hour while the historical usage retrieval component is triggered every month to extract the usage of the last months that is used as a historical usage baseline of the coming month. The output of the historical/hourly usage retrieval component is transferred to a local machine for time series processing.

User location profile retrieval: We build the user location profile retrieval using native Hadoop Map-reduce. The location profile retrieval component is triggered every month to construct the latest location profile that is used for the following month. The user location profile is stored on the Hadoop HDFS file system as the usage aggregation components need the information. (Figure 6).

Time series processing component: This component is located locally on a single node in the Hadoop cluster. The component consists of light-weighted modules such as the time decomposition module, the anomaly detection module, false-alarm removal module etc.

6. EVALUATION SETUP

Obtaining ground truth about service disruptions is inherently difficult. (We do, however, evaluate ABSENCE against known service disruptions in Section 8.1.) To allow a systematic evaluation of our approach we introduced synthetic service disruptions into data obtained from a US mobile provider.

6.1 Data overview

We used CDR data collected in a large US mobile network from July 2014 to December 2014 in our evaluation. We used all 6 months worth of data to build up the historical data used in ABSENCE. With 6 months of historical data and the weekly seasonal model, we maintain $6*4=24$ usage histories for each of the nodes in our geographical hierarchy to use as reference points for anomaly detection in current usage data. This amount of historical usage data is sufficient to maintain acceptable confidence intervals for the time decomposition algorithm. The total amount of data used in our evaluation is 45 TB. The volume of collected CDR data varies from 15-20 GB per hour, depending on user activity, with the monthly volume of 7-8 TB. For the synthetic evaluations presented below, we excluded a week's worth of data from the 6 months data set and used those days as "current usage data" in which the synthetic failures were introduced.

6.2 Synthetic service disruptions

To allow for a systematic evaluation of ABSENCE we emulate service disruptions due to both the network and the device failures. ABSENCE performs service disruption detection by identifying changes in the expected usage data. As such, to emulate both network and device failures, our approach is to *remove* the corresponding data (i.e., data that would disappear if the failure had occurred) from the CDR data for each synthetic scenario.

We mimic *network* failures at the granularity of a base station, i.e., when a base station is down, the service at that base station is lost. In the CDR data, every call record is associated with a list of base stations that served the call, i.e., the "serving list". When we emulate the failure of a base station, we use the serving list to remove call records associated with the base station in question.

We similarly emulate *device* failures by removing all call records associated with the emulated device failure. For example, a firmware bug could affect all devices from one manufacturer after a firmware update and prevent users from making calls even when the network is healthy. To introduce this type of failures, we remove call records with the device make/model associated with the emulated failure.

To introduce different failure scenarios for our evaluation, we combine the basic network and device failures described above with *geographic* information at different granularities and a *severity* factor to be applied. We vary the severity of a failure by failing different numbers of base stations or devices in the chosen aggregation, e.g., 10% of base stations in a ZIP for less severe failures and 100% of base stations in a ZIP for large outages.

6.3 Sensitivity to failure impact

ABSENCE detects anomalies based on variations in the expected normal usage patterns. A key question to answer with our evaluation of this approach is what degree of impact ABSENCE will be able to detect. To answer this question we investigated two factors: (i) failure impact ratio and (ii) absolute impact. Failure impact ratio is defined as the ratio of the total amount of usage reduction during the failure over the total amount of a normal usage, or

$$impact\ ratio = \frac{total\ usage\ reduction}{total\ normal\ usage}.$$

Absolute impact is defined as the total usage reduction during an injected or detected event. For example, if during a 5-hour event, 4,000 out of 5,000 calls were lost, the absolute impact is 4,000 and the impact ratio is 80%. The smaller the impact ratio is, the more challenging it is for an anomaly detection system to identify it. The larger the absolute impact is, the more important it is for operators to pay attention to it. We use both metrics in evaluating ABSENCE regarding the sensitivity and performance in detecting anomalies.

6.4 Failure scenarios

We consider two different geographical aggregation levels: city and ZIP-code area. In term of devices, we consider two popular mobile-device manufacturers namely A and B and two popular mobile-device models, A-1 and B-1. Combinations of the two geographical aggregations and two specific device types allow a variety of test scenarios: city (e.g., all phones in Los Angeles), city+device make (e.g., all A phones in Los Angeles), city + device model (e.g., all A-1 phones in Los Angeles), ZIP code (all phones in ZIP code 07921), ZIP code + device make (e.g., all B phones in ZIP code 07921), etc. To come up with our final failure scenarios we consider three additional attributes: the type of service, the time, duration and the severity of the event (i.e., failure impact ratio).

In order to thoroughly evaluate ABSENCE we generate failure scenarios randomly based on different aspects we want to evaluate. Table 2 shows all aspects and the evaluation values from which we randomly selected to make up our failure scenarios. The table also shows an example scenario for each aspect.

We randomly chose 100 ZIP codes and 10 cities for the geographical aggregations to evaluate. We generated failures with different impacts by varying the amount of failed base stations when generating the failures, i.e., the impact ratio. There are 11 impact ranges each is 5% of impact ratio wide, i.e., [0%-5%], [5%-10%] etc.

We randomly picked 100 failures (i.e., 100 samples) for each impact range, e.g., we picked 100 failures that have [10%-15%] of impact, 100 failures that have [15%-20%] of impact, etc., until all the impact ranges are covered. Note that for each impact range the randomly generated failures should be uniformly distributed across attributes. For example, 100 failures in the (ZIP code + device make) aggregation could happen either to all A devices or B devices and could last for 1,2,3 or 6 hours etc. This way, the set of generated failures should uniformly cover many failure types across attributes and therefore ABSENCE would have a set of diverse failure scenarios to evaluate against.

After generating this "pool" of failure scenarios and using ABSENCE to detect them, we gather the results and break it down into different dimensions based on the aspects in Table 2. In this manner we generated a total of 11,000 synthetic failures across our evaluation space. We present our evaluation results in Section 7.

6.5 Evaluation Metrics

We evaluated ABSENCE using two metrics: detection rate (%) and loss ratio (%). Detection rate is defined as the ratio of correctly detected failures (true positive,TP) over the total amount of introduced failures (true positive,TP + false negative,FN).

$$Detection\ Rate(\%) = \frac{TP}{TP+FN}.$$

Detection rate quantifies how effective ABSENCE is. The higher the detection rate, the more effective ABSENCE.

Aspect	Evaluated values	Example of a failure
Geographical aggregation	100 ZIPs, 10 cities	All devices in L.A. fails
Device make	A, B	All A devices in ZIP 07921 fails
Device model	A-1, B-1	All A-1 devices in L.A. fails
Service	Voice, LTE	All devices in ZIP 07921 can't make calls or can't access the Internet
Start time	quiet period (06:00 UTC), busy period (20:00 UTC)	All devices in ZIP 07921 fails starting from 20:00 UTC
Duration	1, 2, 3, 6, 12 hours for busy; 8,10,12 hours for quiet	Voice service in ZIP 07921 outages for 8 hours starting from 06:00 UTC
Severity impact	0% to 55% of the total usage	A failure that causes 20% reduction of the normal usage in ZIP 07921

Table 2: Aspects and evaluated values of generated failures

Loss ratio is defined as the ratio of the total net-loss *until detection* over the total amount of normal usage during the failure.

$$Loss\ Ratio(\%) = \frac{netloss\ until\ detection}{normal\ usage\ during\ failure}.$$

For example, if the normal usage during a failure is 5,000 and ABSENCE detects the failure when 1,000 calls get dropped then the loss ratio will be 1/5 (20%). In short, for a long lasting failure, the lower the loss ratio means the faster ABSENCE detected the failure.

7. EVALUATION RESULTS

7.1 Variable-scale decomposition

The variable-scale decomposition technique is designed to ensure that ABSENCE uses sufficient usage data to enable accurate detection. To evaluate the effectiveness of this approach we evaluated ABSENCE with and without the variable-scale decomposition technique. We focused our evaluation on ZIP code level aggregations during quiet hours.We introduced failures starting from 03:00 UTC (typically the start of network quiet time) and lasting for 8 hours for voice service. The impact of the failures is in the range of 5% to 100%. We ran the two decomposition techniques (with and without the variable-scale mechanism) over the same set of failures and compare the detection rate of the two.

As shown in Figure 7a, the variable-scale decomposition technique improves the detection rate during the low usage period by *8-10x*, e.g., ABSENCE detected 91% as opposed to 10% of failures that affects more than 2,000 calls in 8 hours respectively with and without the technique.

7.2 Synthetic Failure Evaluation

Overall results. As shown in Table 3 first row, out of 11,000 introduced failures for both Voice and LTE service, ABSENCE was able to detect 9,676 with a detection rate of 88.0%. For failures with an impact larger than 10%, ABSENCE was able to detect 97.7% (8,064 out of 8,254 failures), and for failures that are larger than 20% of impact, ABSENCE detected 99.0% (6,189 out of 6,254 failures).

Table 3 breaks down the detection rate by different aggregations. Overall ABSENCE detected 98% of failures that have more than 10% impact across different aggregations from large aggregations (e.g., city level - all users in LA) to smaller aggregations (e.g., (ZIP code + make) level - all Device A or B devices in a ZIP code). Next, we look into different factors (i.e., failure impact ratio, absolute impact) that affects the detection rate of ABSENCE. With each factor, we also break down the results into different aggregations.

Failure impact ratio and detection rate. We would like to understand the effectiveness of ABSENCE as a function of the severity of failures, i.e., the impact ratio, for different failure scenarios.

(i) Overall results: Figure 7b shows the overall detection rate as a function of the impact ratio of failures across all aggregations and service types. ABSENCE was able to detect 96% of failures that have a 15%-20% of impact across all aggregations, services and device types. For outages (with 50% of impact or more severe), ABSENCE detected 100% of them. For failures that are less severe (i.e., smaller than 10% of impact) ABSENCE detected 20%-67% of them.

(ii) Failures at different aggregation levels: Figure 7c shows ABSENCE's detection rate *for failures with small impact* at different aggregation levels: city, city + device make, city + device model, ZIP code, ZIP code + device make. Overall, for failures with more than 15% of impact, ABSENCE is equally effective across aggregations with a detection rate of 94% or better. This trend continues for failures with higher impact and we omit the results due to space constraints. For failures with lower impact (0%-5% and 5%-10%), ABSENCE's detection rate reduces to between 5% and 80%. For those failures, ABSENCE was slightly more effective for large aggregations such as *city* and *city + device make*.

(iii) Failures happen to different service types: Figure 7d shows ABSENCE's detection rate for failures occurring in LTE and voice services. For failures with more than 15% of impact, ABSENCE's detection rate is high for both voice and LTE (i.e., around 97%). For less severe failures (i.e., less than 15% of impact), ABSENCE detected failures to LTE service slightly better than voice service.

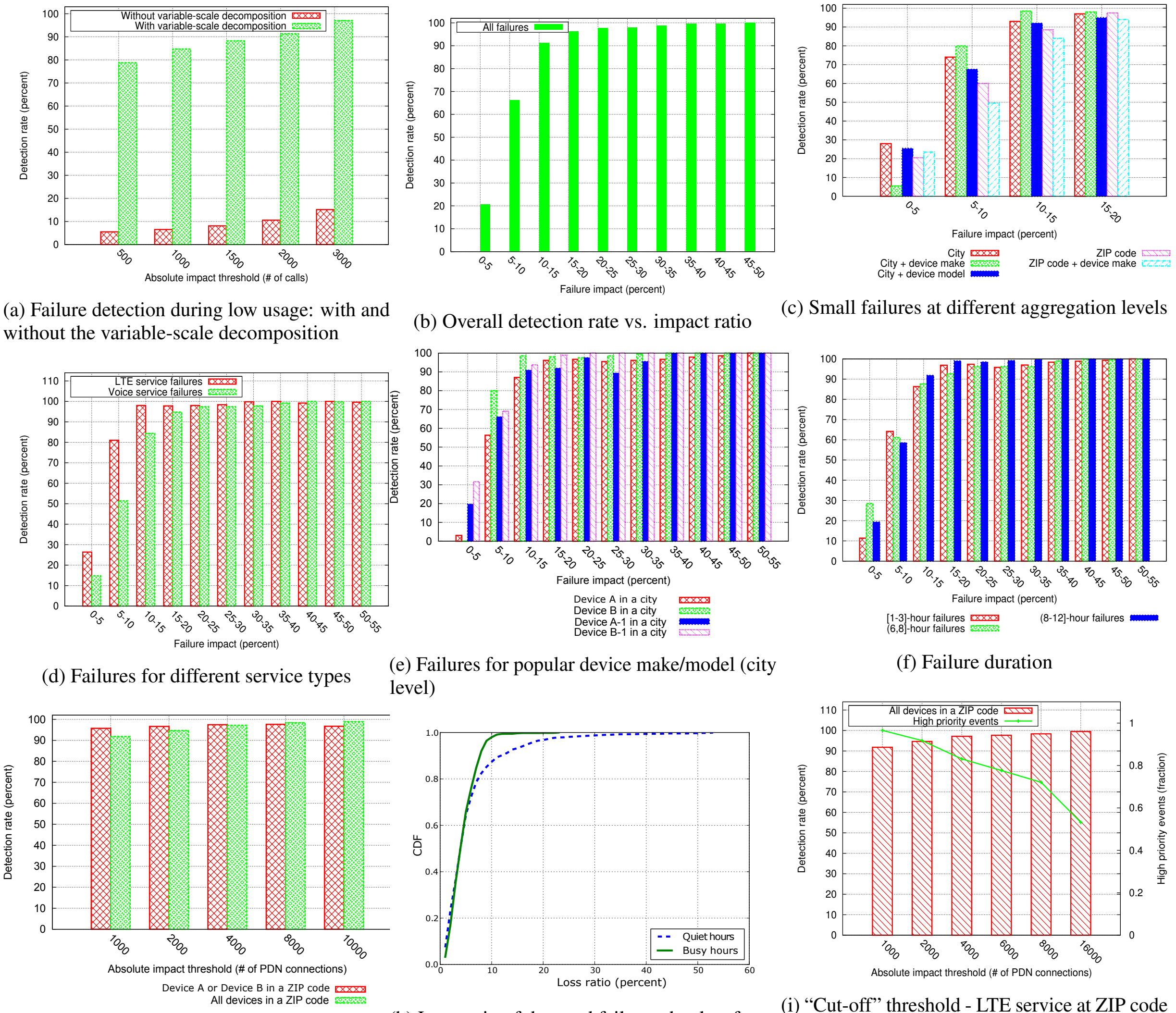
(iv) Failures happen to different mobile device types: Figure 7e shows ABSENCE's detection rate for failures associated with two popular device makes (A and B) and two popular phone models (A-1 and B-1) at city level. ABSENCE detected around 94% of failures with 15%-20% of impact associated with those mobile device makes and models.

(v) Failures break down by duration: Figure 7f shows ABSENCE's detection rate for failures with different durations. Overall ABSENCE is equally sensitive across durations of the failures: it detected about 95% of failures with $15-20\%$ of impact for both short and long-lasting failures. For failures with 0%-10% impact, ABSENCE detected 10%-63% of them.

Absolute impact and detection rate. To understand how effective ABSENCE is in detecting failures ranked by the absolute size, we looked into ABSENCE's detection rate as a function of the absolute impact of the failures. Figure 7g shows that for LTE ABSENCE detected 94% of failures that cause more than 2,000 PDN connections to be dropped at the ZIP code level and 96% of failures at the (ZIP code + device make) level. We omit similar results for voice service because of space constraints.

Loss ratio of detected failures. To understand how quickly ABSENCE can detect long-lasting failures, we obtained the loss ratio (Section 6) of detected failures that last at least 6 hours for all services across different aggregations. We are interested particularly in long-lasting failures because those failures often cause larger impact and early detections mean lower impact.

Figure 7h shows the CDF of the loss ratio of detected failures during quiet hours (i.e., low usage period) and busy hours (i.e., peak usage period). In general, during busy hours 98% of the



(a) Failure detection during low usage: with and without the variable-scale decomposition

(b) Overall detection rate vs. impact ratio

(c) Small failures at different aggregation levels

(d) Failures for different service types

(e) Failures for popular device make/model (city level)

(f) Failure duration

(g) Absolute impact - LTE service

(h) Loss ratio of detected failures that last for at least 6 hours

(i) "Cut-off" threshold - LTE service at ZIP code level

Figure 7: Detection rates

Break down	**Aggregation**	**All failures**	**≥ 10% of impact**	**≥ 20% of impact**
Total	All aggregations	88.0%	97.7%	99.0%
Geographical break down	City	89.6%	98.1%	99.0%
	Zip code	87.5%	97.8%	99.4%
Service break down	Voice	85.2%	96.5%	98.7%
	LTE	90.7%	98.9%	99.3%
Geographical + device break down	City + device make (e.g., A devices in LA)	88.9%	99.0%	99.3%
	City + device model (e.g., A-1 devices in LA)	88.3%	97.3%	98.6%
	ZIP code + device make (e.g, A-1 devices in 07921)	85.5%	96.3%	98.6%

Table 3: Overall results break down by different aggregations

failures were detected when there were less than 10% of the total loss happens (i.e., if the issue is fixed at the detection time, 90% of the usage would be recovered for 98% of the failures). During quiet hours, 90% of the failures were detected when there were less than 10% of loss. Given that the detected failures are at least 6 hours, 10% loss suggests that the failures are mostly detected right after the first hour of the failures.

Impact based event prioritization. Because of resource constraints, in an operational setting providers typically need to prioritize the events that they investigate. Because it is inherently usage based, ABSENCE lends itself to an "operational knob" that operators can tune to distinguish large impact events from the small impact ones, so that they can rapidly respond to more severe conditions. Here we evaluate the tradeoff through such a knob, in the form of a "cut-off" threshold – the threshold above which events are defined as of high priority – between the number of high priority events and the detection rate (rate of high impact events to be included in the high priority list).

Figure 7i shows the detection rate (Y1-axis) and the fraction of synthetic events that are of high priority (Y2-axis) as functions to the threshold for the ZIP code level. We observe that if we only focus on events that have an absolute impact of over 4,000 records, ABSENCE achieves the detection rate of 97% among the 82% of the events.

Implications: Our evaluation of ABSENCE shows that: (i) ABSENCE has a very high detection rate across all scenarios for failures with medium to high impact (i.e., above 15%). (ii) ABSENCE can detect failures relatively quickly (i.e., has a low loss ratio), thus reducing the impact of failures. (iii) Prioritizing events based on the absolute number of missing records provides a simple operational knob that enables operators to tune the number of high priority events generated by ABSENCE. These results suggest the practical feasibility of using ABSENCE to perform service disruption detection at the scale of modern mobile networks.

8. OPERATIONAL VALIDATION

In this section, we validate service impacting events detected in the operational data via ABSENCE with known historical service outages. Specifically, we use events that resulted in anomalous volumes in customer care calls as our known customer impacting network events. Customers can call into customer care centers to report service issues and to discuss other concerns. The vast majority of customer calls relate to individual customer concerns - they may be the result of individual customer device issues or user errors, for example. However, in some situations, these customer calls may be the result of a broader service impacting event. Network, device type or application disruptions can thus result in a spike in the number of customer complaints. These events - spikes in customer care calls - are captured in a database along with their associated underlying network/device/application root cause, and are used here to provide a source of ground truth of service disruptions that we use to compare with those detected by ABSENCE.

We first attempt to validate service disruptions detected by ABSENCE with customer complaint events over a corresponding time period. We then investigate a number of specific use cases in more detail to verify ABSENCE functionality.

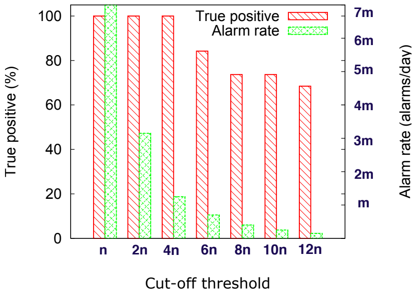


Figure 9: Alarm rate and true positive of ABSENCE

8.1 Comparison with customer complains

From the customer care database, we can extract customer complaint events at a market level, i.e., which market the event is in and the event's start/end time. We obtained a list of a 19 such events from the customer care database that happened to Voice and LTE data services. Operations had confirmed that each of these was a true service disruption. We then attempted to detect these with ABSENCE using CDR aggregates for the corresponding dates.

We ran ABSENCE to get a list of detected events at the ZIP code level and compared the detected events with the 19 customer complaint events mentioned above. Customer complaint events were aggregated at the market level (i.e., a market typically consists of several geographically close cities), while we were detecting events at the ZIP code level. As a result, before performing the comparison, we mapped the ZIP code of ABSENCE detected events to the corresponding market. We considered a *match* between an event detected by ABSENCE and a customer complaint event if the two events are spatially and temporally matching. Using this approach, ABSENCE was able to detect *all* 19 customer complaint events. We noted that ABSENCE also detected possible service disruptions that are not included in the customer care database. Due to lack of ground truth available in this study, it is challenging to investigate these further.

8.2 Alarm rate and true positive rate

ABSENCE detected events that are not included in the customer care database. Due to lack of ground truth it is challenging to determine whether those events are false positives. However, to make ABSENCE practical, the number of events per day (i.e., alarm rate) should be reasonable for an operations team to handle while the true positive rate should be maintained. To adjust the alarm rate, ABSENCE uses different cut-off thresholds (i.e., amount of usage impacted per hour) to filter out events with relatively small impact, i.e., if the impact of an event is smaller than a certain threshold, the event will not trigger an alarm. We used the 19 events in the customer database as the ground truth for the true positive rate and varied the cut-off threshold to observe the alarm rate of ABSENCE. Figure 9 shows the alarm rate and the true positive rate as a function of the cut-off thresholds. As we can see, both the alarm rate and the true positive decrease as the cut-off threshold increases. To protect proprietary information, we show the alarm rate and the cut-off threshold as relative numbers. If ABSENCE only triggers alarms for events which impact greater than 4*n* calls/PDN connections per hour, then the operations team will need to investigate around *m* such events per day and ABSENCE detects all of the 19 events above (i.e., 100% true positive). We confirmed that *m* is manageable by Operators and thus ABSENCE is practical in an operational environment.

8.3 Use cases

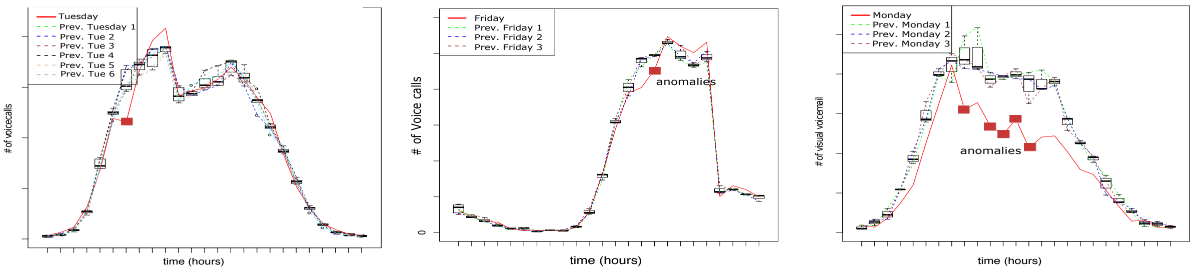
In this section we explore a number of specific use cases where ABSENCE was able to detect anomalies that also showed up in the customer care database. In this section, ABSENCE used 4*n* as the cut-off threshold for the number of calls or PDN connection per hour.

(i) Voice service in a large metropolitan area:

This failure was on the voice service in several ZIP codes of a large metropolitan area. Users in these areas were not able to receive calls from landline devices. The event started at 16:00 UTC on a given Tuesday according to the customer ticket data. Figure 8a shows the time series in UTC of the historical usage data for voice services of one of the affected ZIP codes (dashed line) for previous Tuesdays and the abnormal usage on the date of the service disruption (red solid line). ABSENCE was able to detect an anomalous event at 16:00 UTC (red point) as the usage falls significantly outside of the range of the normal historical usages.

(ii) Voice service in a large metropolitan area:

This failure was on the voice service in several ZIP code areas in another large metropolitan area. The failure was first evident in customer ticket volume at 12:00 UTC on a Friday according to the customer complaints and the reports by the operator. Figure 8b shows the historical usage (dashed lines) of previous Fridays and the usage of the day that the failure occurred (red solid line) for one of the affected ZIP code areas. As we can see, there are drops in the usage on the day of the failure and ABSENCE can detect the anomaly at 14:00 UTC (red point) – two hours after it commenced.



(a) Voice service anomaly: metro area 1

(b) Voice service anomaly: metro area 2

(c) Voicemail service anomaly

Figure 8: Use cases

Note that the sudden drop later in the day is due to the change in the number of users in the area according to the change of the zip-code profile.

(iii) Voice mail:

In this use case the failure happens to a much larger area (i.e, all ZIP codes in a large metropolitan area). This failure started around 18:00 UTC on a Monday and was associated with one of the operator's Voicemail services - a data service that is available to only a subset of user devices. This use case is an example of the failure of a particular service that impacted only particular device types within a given area. ABSENCE detected a series of anomalies happening to devices in the metropolitan area commencing at 18:00 UTC (when the failure itself commenced) (Figure 8c solid red line).

9. DISCUSSIONS

Special events: Special events and holidays may affect users' usage and mobility pattern, which can be challenging for a passive monitoring approach. For example, people make less calls on holidays, or during a sport event. Correlated usage reduction in large scale can cause ABSENCE to generate false positives. If users' service consumption does not change but their mobility pattern changes (e.g., gathering at a stadium and use the phones as normal), ABSENCE will not generate false positives because the location profile will handle this and their usages are counted toward the profile location regardless of where they are.

Metrics to use: ABSENCE uses number of calls and number of PDN connections as the metric for detecting service disruptions. Those two metrics could quantify users' experience in most cases, e.g., if a base station fails and users could not attach or make calls/create PDN connections. Potentially, there are other metrics that could be used such as duration of calls or number of Bytes downloaded/uploaded. These metrics would capture other types of failures. For example, a failure at the routing system in the network may cause voice calls to be routed to voice mail instead of reaching the callee. In this case the total duration of calls would be a better indicator of a potential service impairment worthy of investigation.

New services in Data network: Network operators may introduce new data services such as Wi-Fi offload. Such new services may affect the usage captured in the CDRs, as load is (deliberately) reduced. Since ABSENCE uses multiple months of historical usage as the baseline, if Wi-Fi offload skews the usage, the baseline will be rebuilt. After a sufficiently long period of training (typically 3-4 weeks if a weekly seasonal model is used), the new baseline will include the Wi-Fi offload and ABSENCE will be able to operate in the new environment.

10. RELATED WORK

There are quite a lot related work in the area of service disruption detection including both commercial systems such as Keynote [2] and Gomez [1] and various efforts by the research community [16, 11, 30, 28]. All of them share two limitations. First, their effectiveness are typically limited by the coverage of deployed probes. Second, they all need to inject unnecessary probing traffic to the system, which could affect legitimate users. In the contrast, ABSENCE detects service disruptions in a non-intrusive (passive) manner by purely depending on the existing traffic from real users.

Our work also relates to various mobile network performance studies. For example measurements from mobile devices have been used to study the performance of mobile networks [20]. Several studies have investigated protocol level performance aspects of mobile networks [14, 22, 19]. While these detailed performance aspects of mobile networks are related to our work, ABSENCE is a network management tool dealing with the operational health of a mobile network. As such ABSENCE is most related to various network operations tools [29, 17, 24, 26, 18]. A framework for network anomalies based on a principal component analysis approach has been proposed in [29]. A performance troubleshooting tool [17] and a service quality assessment mechanism [24] for IPTV networks have been developed. Service anomaly detection [26] and troubleshooting tools [18] have been developed for ISP networks. In contrast ABSENCE is focused on service disruption detection in mobile networks and deal with the specific mobility and scalability challenges of that environment. Production tools such as [7] is installed on user devices to collect users' service experience yet they are not available on most Samsung and iPhone popular models due to privacy issues. ABSENCE in contrast works for all device models without installing any software on the device.

11. CONCLUSION

We presented our work on ABSENCE, a service disruption detection system for mobile networks. ABSENCE makes use of customer usage data, in the form of aggregated and anonymized call detail records, to derive historical usage patterns for groups of customers. Appropriate selection of these groups results in stable and accurate predictions of usage patterns, allowing ABSENCE to detect deviations as possible service disruptions. We presented a data driven exploration of the design space. We performed a systematic evaluation of ABSENCE by introducing synthetic failures in data from an operational mobile network and compared ABSENCE's detection results with known ground truth events from the mobile network. ABSENCE is currently operating in a pre-production environment. Our future plans include integration of ABSENCE with the operator's production environment and fine tuning the parameters to improve the accuracy and utility of our approach in an operational setting.

12. ACKNOWLEDGMENTS

We would like to thank our anonymous reviewers and shepherd for their valuable comments. This research was supported in part by the National Science Foundation under grants 1305384 and 1302688.

13. REFERENCES

[1] Gomez, inc. website. http://www.gomez.com/.

[2] Keynote systems, inc. website. http://www.keynote.com/.

[3] 3GPP. Telecommunication management; Key Performance Indicators (KPI) for Evolved Universal Terrestrial Radio Access Network (E-UTRAN): Definitions. http://www.3gpp.org/DynaReport/32450.htm.

[4] Ahuja, S., Ramasubramanian, S., and Krunz, M. SRLG Failure Localization in All-Optical Networks Using Monitoring Cycles and Paths. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE* (April 2008), pp. –.

[5] Amrutkar, C., Hiltunen, M., Jim, T., Joshi, K., Spatscheck, O., Traynor, P., and Venkataraman, S. Why is my smartphone slow? on the fly diagnosis of underperformance on the mobile internet. In *Dependable Systems and Networks (DSN), 2013 43rd Annual IEEE/IFIP International Conference on* (June 2013), pp. 1–8.

[6] Booker, G., Torres, J., Guikema, S., Sprintson, A., and Brumbelow, K. Estimating cellular network performance during hurricanes. *Reliability Engineering & System Safety 95*, 4 (2010), 337–344.

[7] CarrierIQ. Carrier IQ, Inc. http://www.carrieriq.com.

[8] Chatfield, C. *The analysis of time series: an introduction*. CRC press, 2004.

[9] Cisco Systems. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019. *White Paper, February* (2015).

[10] Dong, W., Ge, Z., and Lee, S. 3G Meets the Internet: Understanding the performance of hierarchical routing in 3G networks. In *Teletraffic Congress (ITC), 2011 23rd International* (Sept 2011), pp. 15–22.

[11] Feamster, N., Andersen, D., Balakrishnan, H., and Kaashoek, M. Measuring the effects of internet path faults on reactive routing. In *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems* (2003), ACM, p. 137.

[12] Gember, A., Akella, A., Pang, J., Varshavsky, A., and Caceres, R. Obtaining in-context measurements of cellular network performance. In *Proceedings of the 2012 ACM conference on Internet measurement conference* (2012), ACM, pp. 287–300.

[13] Goyal, M., Ramakrishnan, K., and chi Feng, W. Achieving faster failure detection in OSPF networks. In *Communications, 2003. ICC '03. IEEE International Conference on* (May 2003), vol. 1, pp. 296–300 vol.1.

[14] Huang, J., Qian, F., Guo, Y., Zhou, Y., Xu, Q., Mao, Z. M., Sen, S., and Spatscheck, O. An in-depth study of LTE: effect of network protocol and application behavior on performance. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM* (2013), ACM, pp. 363–374.

[15] Kalmanek, C. R., Misra, S., and Yang, Y. R. *Guide to reliable internet services and applications*. Springer, 2010.

[16] Kompella, R., Yates, J., Greenberg, A., and Snoeren, A. Detection and Localization of Network Black Holes. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE* (May 2007), pp. 2180–2188.

[17] Mahimkar, A., Ge, Z., Shaikh, A., Wang, J., Yates, J., Zhang, Y., and Zhao, Q. Towards Automated Performance Diagnosis in a Large IPTV Network. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication* (New York, NY, USA, 2009), SIGCOMM '09, ACM, pp. 231–242.

[18] Mahimkar, A., Yates, J., Zhang, Y., Shaikh, A., Wang, J., Ge, Z., and Ee, C. T. Troubleshooting chronic conditions in large IP networks. In *Proceedings of the 2008 ACM CoNEXT Conference* (2008), ACM, p. 2.

[19] Nguyen, B., Banerjee, A., Gopalakrishnan, V., Kasera, S., Lee, S., Shaikh, A., and Van der Merwe, J. Towards understanding TCP performance on LTE/EPC mobile networks. In *Proceedings of the 4th workshop on All things cellular: operations, applications, & challenges* (2014), ACM, pp. 41–46.

[20] Nikravesh, A., Choffnes, D. R., Katz-Bassett, E., Mao, Z. M., and Welsh, M. Mobile Network Performance from User Devices: A Longitudinal, Multidimensional Analysis. In *Passive and Active Measurement* (2014), Springer, pp. 12–22.

[21] Nikravesh, A., Choffnes, D. R., Katz-Bassett, E., Mao, Z. M., and Welsh, M. Mobile network performance from user devices: A longitudinal, multidimensional analysis. In *Passive and Active Measurement* (2014), Springer, pp. 12–22.

[22] Qian, F., Sen, S., and Spatscheck, O. Silent TCP connection closure for cellular networks. In *CoNEXT* (2013), pp. 211–216.

[23] Roughan, M., Greenberg, A., Kalmanek, C., Rumsewicz, M., Yates, J., and Zhang, Y. Experience in measuring Internet backbone traffic variability: Models, metrics, measurements and meaning. In *Proceedings of the International Teletraffic Congress (ITC-18)* (2003).

[24] Song, H. H., Ge, Z., Mahimkar, A., Wang, J., Yates, J., Zhang, Y., Basso, A., and Chen, M. Q-score: Proactive Service Quality Assessment in a Large IPTV System. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference* (2011), IMC '11, ACM.

[25] Xu, Y., Wang, Z., Leong, W. K., and Leong, B. An end-to-end measurement study of modern cellular data networks. In *Passive and Active Measurement* (2014), Springer, pp. 34–45.

[26] Yan, H., Flavel, A., Ge, Z., Gerber, A., Massey, D., Papadopoulos, C., Shah, H., and Yates, J. Argus: End-to-end service anomaly detection and localization from an ISP's point of view. In *INFOCOM* (2012), IEEE.

[27] Yan, H., Ge, Z., Osinski, M., and Yates, J. When Cell Towers Fail: Quantifying the Customer Impact. http://www.research.att.com/articles/featured_stories/2013_03/201306_tower-outage-analyzer.html.

[28] Zhang, M., Zhang, C., Pai, V., Peterson, L., and Wang, R. PlanetSeer: Internet path failure monitoring and characterization in wide-area services. In *Proc. USENIX OSDI* (2004).

[29] Zhang, Y., Ge, Z., Greenberg, A., and Roughan, M. Network Anomography. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement* (Berkeley, CA, USA, 2005), IMC '05, USENIX Association, pp. 30–30.

[30] Zhang, Y., Mao, Z., and Zhang, M. Effective diagnosis of routing disruptions from end systems. In *Proceedings of USENIX NSDI*, vol. 8.