# When the Network Crumbles: An Empirical Study of Cloud Network Failures and their Impact on Services

Rahul Potharaju
Purdue University
rpothara@purdue.edu

Navendu Jain
Microsoft Research
navendu@microsoft.com

## Abstract

The growing demand for always-on and low-latency cloud services is driving the creation of globally distributed datacenters. A major factor affecting service availability is reliability of the network, both inside the datacenters and wide-area links connecting them. While several research efforts focus on building scale-out datacenter networks, little has been reported on real network failures and how they impact geo-distributed services. This paper makes one of the first attempts to characterize *intra-datacenter* and *inter-datacenter* network failures from a service perspective. We describe a large-scale study analyzing and correlating failure events over three years across multiple datacenters and thousands of network elements such as Access routers, Aggregation switches, Top-of-Rack switches, and long-haul links. Our study reveals several important findings on (a) the availability of network domains, (b) root causes, (c) service impact, (d) effectiveness of repairs, and (e) modeling failures. Finally, we outline steps based on existing network mechanisms to improve service availability.

## Categories and Subject Descriptors

C.2.1 **[Computer Communication Networks]:** Network Architecture and Design; D.2.3 **[Computer Communication Networks]:** Network Operations

## Keywords

Datacenters; Network reliability; Inter-datacenter links; Cloud services

*SoCC'13*, 1–3 Oct. 2013, Santa Clara, California, USA.
ACM 978-1-4503-2428-1.
http://dx.doi.org/10.1145/2523616.2523638

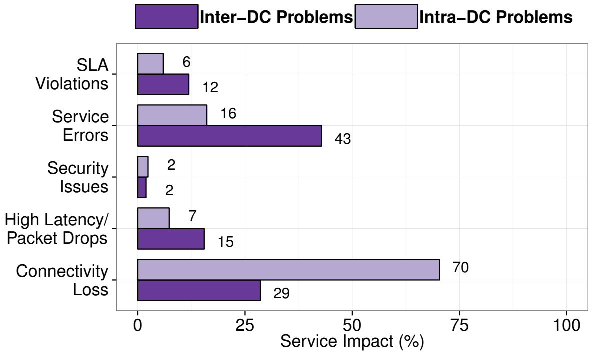


**Figure 1:** The top-5 categories of service impact observed from the high severity incidents attributed to Intra-DC network devices (Layer-3 routers and Layer-2 switches) and Inter-DC network links. Connectivity loss problems (70%) and Service Errors (43%) dominate the impact due to Intra-DC and Inter-DC network problems, respectively.

## 1 Introduction

Cloud services are growing rapidly to provide a fast-response and always-on experience to end users. Reliability is critically important for these services as failures not only hurt site availability and revenue, but also risk data loss. For instance, Dropbox experienced two recent widespread outages [13, 47] which prevented its users from synchronizing files or accessing its site. In another instance, the hurricane Sandy led to flooding of many datacenters in NYC causing service outages [15], and failures of many trans-atlantic fiber links peering from NYC significantly degrading capacity [50]. In 2011, the entire US East region of Amazon became unavailable due to a faulty fail-over during maintenance [3] impacting several popular services such as Dropbox, Foursquare, Instagram, Quora and Reddit.

To increase service availability in a cost-effective manner, cloud providers are deploying their services across geo-distributed datacenters [23], and building their networks based on a scale-out design using inexpensive commodity hardware [17, 34, 36]. However, as the number of devices and links in a datacenter grows, failures become the norm rather than the exception. Making things worse, the network infrastructure also comprises long-haul, inter-datacenter links to synchronize and replicate user data and application state [2]. These links comprise a variety of components such as cables, optical adapters, and complex software protocol stacks, whose failures can lead to reduced bandwidth capacity, stale data, or even service outages.

Unfortunately, despite their practical significance, little is known about how cloud network failures impact services. The research literature offers several real-world studies on failures of disk and storage systems [4, 19, 38, 43], DRAM [44], personal computers [35], and errors in software configuration [51], but they do not consider network failures. Recent studies [46, 48] including our own [16, 39, 40] study failures of network switches and middleboxes, *but they neither analyze their impact on cloud services nor do they examine inter-datacenter network failures*; we compare to prior work in detail in §9.

## 1.1 Motivation

**Types of Service Impact:** Figure 1 shows the top-5 categories of service impact caused by network failures based on our analysis of high severity incidents spanning five years (2008-13) in a cloud provider comprising dozens of datacenters; an incident is high severity if it causes high customer or business impact. The network elements comprise both Intra-DC Layer-3 and Layer-2 devices (Access routers, Aggregation switches and Top-of-Rack switches) and Inter-DC links. We observe that loss of connectivity and service errors (e.g., replication problems leading to stale data) dominate the service impact. Further, we observe that Inter-DC network failures are caused due to link flapping (36%), high link utilization (29%) and unplanned changes (6%). In comparison, the Intra-DC network failures are dominated by connectivity errors (64%-78%), hardware failures (20%-73%) and software problems (7%-24%) across Layer-3 and Layer-2 devices.

**Categories of Impacted Services:** Our analysis of the high severity incidents revealed that network failures impact a broad range of services. The Intra-DC network failures mainly affected messaging (e.g., email, IM services, SMS) services in 38.9% of the incidents. SaaS applications (e.g., web hosting, CDN, data analytics) were affected in 32% of the incidents equally due to Intra- and Inter-DC problems where the geo-distributed services were disrupted for up to several hours. Thus, understanding network failures at both the intra- and inter-datacenter level is important to deliver high availability for cloud services.

## 1.2 Our Contributions

In this paper, we perform one of the first characteristic studies of cloud network failures from a service perspective. Our study using real-world data focuses on understanding the failure modes and correlation of network failure logs and how they impact cloud services. Specifically, we aim to answer the the following questions:

Q1 **Network stamp availability of a service**: What are the failure characteristics of the network stamp (set of all network elements rooted at a pair of Layer-3 Access routers) of a service inside a datacenter as well as across geo-distributed datacenters? How effective are redundancy mechanisms in handling intra- and inter-datacenter network failures? How many independent network stamps are needed to meet an uptime service-level-agreement (SLA) of a cloud service?

Q2 **Causes of network failures**: What are the main root causes of network failures?

Q3 **Failure Modeling**: How to model failures of network components? Are failures recurrent? Are they bursty? Are device-level repairs effective?

Q4 **Capacity vs. Availability**: For commodity Layer-2 switches, how do their port capacity in terms of connected devices affect their availability in operation?

This paper analyzes the failure characteristics of network stamp of a service comprising Access routers, Aggregation switches, Top-of-Rack and inter-datacenter links, based on three years' (2010-13) worth of network event logs collected in a cloud provider network across thousands of devices spanning dozens of datacenters. Our data covers a wide range of network data sources, including syslog and SNMP alerts, network trouble tickets, maintenance tracking and revision control system, and traffic carried by links.

Our study reveals many key findings that can provide useful guidelines to improve network reliability for geo-distributed services. Our major findings are as follows:

1. Network failures cause significant impact to cloud services, dominated by connectivity loss problems (70%) and service errors (43%) due to Intra-DC and Inter-DC network problems, respectively.
2. The number of independent network stamps for a desired uptime SLA of 3 9's (maximum 8.76 hours of downtime per year) is three and for 4 9's (maximum 52 minutes of downtime per year) is four.
3. Network redundancy is least effective at the Access router-Aggregation switch layer and is most effective at the Inter-datacenter level.
4. Network device failures are not memoryless and exhibit the "few bad apples" effect.

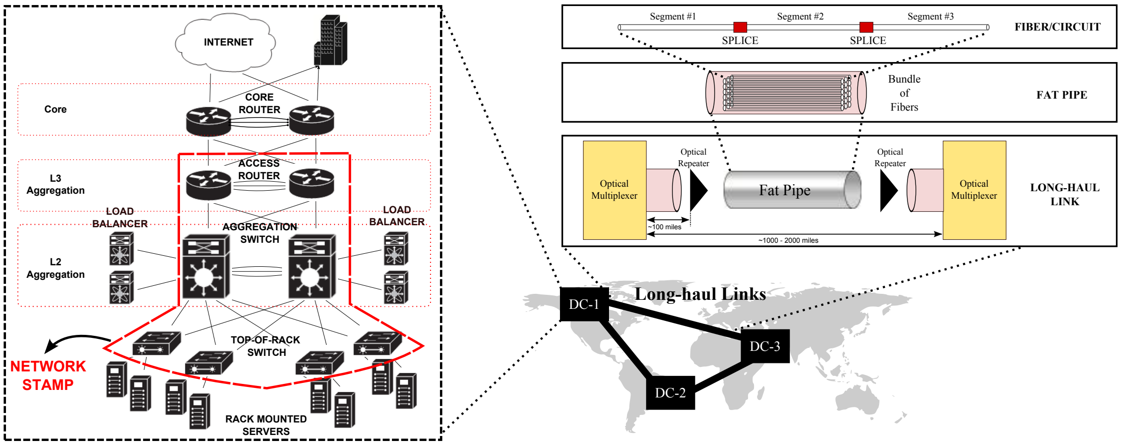


**Figure 2:** Example of a datacenter network architecture. Long-haul links (typically, optical fibers) connect geo-graphically distributed datacenters and can span thousands of miles.

5. Layer-2 Aggregation switches exhibit high availability when up to half of their port capacity is utilized in terms of Top-of-Rack switch count. However, the availability quickly decreases as the Top-of-Rack switch count increases.
6. Fiber length in Inter-DC links is not correlated with the number of failures, and links with high utilization exhibit 2x-3x higher downtime than expected.
7. Top-of-Rack switches exhibit an increase in probability of a successive device failure after repair, while this probability decreases for Aggregation switches and Access routers.

Note that there are other resiliency mechanisms deployed at compute and storage layers of a service which are complementary to the network layer studied in this paper. While these mechanisms may be able to mask the impact of some of the network failures, they require service operators to carefully balance the trade-off between consistency and availability particularly under network partitions, due to the famous CAP dilemma [6]. We plan to study the overall impact of redundancy mechanisms at the storage, compute, and network layers in future work.

# 2 Background

In this section we present an overview of the datacenter network architecture and long-haul links connecting geo-distributed datacenters.

## 2.1 Intra-Datacenter Topology

A datacenter network is typically set up as a multi-root spanning tree topology comprising different types of devices such as routers, switches, load balancers, and firewalls. Figure 2 illustrates an example topology of a datacenter network based on the functional separation of Layer-2 (trunking, VLANs, etc.) and Layer-3 (routing) responsibilities. Top-of-Rack (ToR) switches connect servers hosting applications via 10/100/1000 Ethernet links, with the uplinks being either 1GE or 10GE ports. The ToRs are connected upstream to Aggregation switches (AGG) which serve as an aggregation point for the Layer-2 traffic. Traffic from AGGs is forwarded to Access routers (ARs) that use Virtual Routing and Forwarding (VRF) to create a virtual, Layer-3 environment for each tenant. The ARs aggregate traffic from up to several thousand servers and route it to core routers that connect to other datacenters and the Internet. To provide fault tolerance, network devices are typically deployed in 1:1 redundancy pairs or larger groups.

A **network stamp** (shown in dashed red in Figure 2) is the set of all network elements that are rooted at a pair of Layer-3 ARs, comprising multiple Layer-2 AGG domains in the underlying subtrees.

## 2.2 Inter-Datacenter Connectivity

Geographically distributed datacenters are connected to each other and to the Internet typically using long-haul WDM (wave division multiplexing) optical transport networks spanning about 3000 miles between two endpoints. WDM is usually operated either in a coarse (CWDM) or dense (DWDM) multiplexing manner. While the former utilizes multiple wavelengths spaced at 20nm and operates in the 1271-1611nm spectrum range, the latter utilizes many wavelengths spaced narrowly at 0.8nm and operates in the 1530-1565nm spectrum range (C-band). Modern coherent receivers use polarization multiplexed quaternary phase shift keying (PM-QPSK) modulation and can achieve 100Gbps transmission on

50GHz ITU channel grid, and a total capacity of 8Tbps in the C-band.

Long-haul fiber resources are scarce, expensive and time consuming to construct as well as to fix as engineers may have to travel to the remote physical location. As shown in Figure 2, unlike traditional telecommunication networks that require a lot of intermediate add/drop points (e.g., optical multiplexers, signal repeaters), inter-dc links are mostly point-to-point fat pipe connections with few intermediate add/drops. Fat pipes contain multiple *segments* spliced together to form an optical *circuit* also known as a long-haul fiber.

# 3 Data Sources and Methodology

In this section we first describe the multiple sources of data collected by network operators, comprising our large-scale dataset spanning three years (July 24, 2010 - June 24, 2013). Second, we describe the key challenges in accurately extracting failures from raw networks events. Finally, we present a systematic methodology based on event processing to address them.

## 3.1 Network Datasets

Our dataset includes multiple sources of network failure data spanning three years logged in monitoring servers of a large cloud provider comprising 100k+ servers and 10k+ Layer-2 and Layer-3 devices across 10+ datacenters. These datacenters host a variety of applications ranging from customer facing ones such as web services, video streaming, data stores, and enterprise applications to data intensive applications such as search indexing and MapReduce jobs.

**Network Event Logs**: Network failures are typically detected from monitoring alarms such as syslog and SNMP traps and tracking the health of each device/link via ping and SNMP polling. These logs contain information about the network element experiencing the event, the event type, the other end- point of this device/link, and a short machine-generated description of the event.

**High Severity Incidents**: To analyze impactful incidents where service outages occur and customers get impacted, operators keep details of each high severity incident. Similar to the trouble ticket data, each high severity incident has a unique ticket identifier and contains both structured and unstructured information which we leverage for problem inference. We use this dataset over a period of five years (2008-13).

**Trouble Tickets**: To track network faults during troubleshooting, a ticketing system is used typically based on the NOC RFC [20]. This system coordinates tasks among network engineers working on an incident. Tickets have a unique identifier and contain both structured information about the failure (such as when and how a failure was discovered) and a diary of steps taken by operators to resolve the problem. Based on the results from our prior study [41], we cannot use the structured information for any problem inference due to their high inaccuracy. Therefore, we leverage NetSieve [41] on the network support tickets to infer root causes for (1) high severity incidents and (2) failure events including maintenance-related network changes.

**Maintenance Data**: To track activities such as device repairs/provisioning, configuration changes, and software upgrades throughout the network, operators use a maintenance tracking and revision control system. It serves as a repository of syslog information and includes comments from network engineers about when and why changes were performed. Before debugging an outage, an engineer checks this repository for on-going maintenance and verifies any recent changes to the device configuration. We also obtained maintenance tracking information for inter-datacenter long-haul links where the operators recorded the expected duration of a fiber or segment to be down. To avoid skewing the failure distributions due to maintenance events, we compute the downtime of devices/links separately for unexpected failures and planned changes.

**Network Traffic Data**: We utilize traffic averages observed every five minutes on network interfaces logged using SNMP [9] polling. Traffic monitoring systems use the MIB [31] format to store the data that includes fields such as the interface type (token ring, ethernet etc.), other end of the interface, interface status (up/down), number of bytes sent/received. We correlate this traffic data with failure events to extract failures impacting network traffic, and to reverse-engineer the topology using active link-level connectivity.

## 3.2 Obtaining Impactful Events

We define a *device failure* as an event that causes a device to be inoperational to carry traffic and a *link failure* as an event that causes a link to be down or that causes excessive packet discards. While these definitions are simple and intuitive, there are several key challenges in utilizing the network event logs for studying device/link failure characteristics:

1. Syslog messages can be significantly noisy with devices logging multiple down notifications even though a device/link is operational, or multiple down and up messages as different events due to flapping
2. Redundant events resulting from two devices (e.g., neighbors) logging notifications for the same event

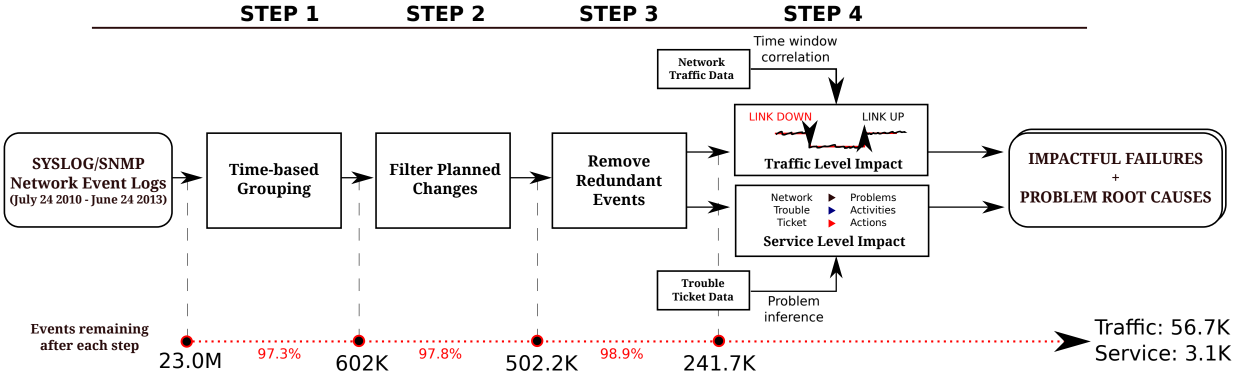


**Figure 3:** A pipeline of event processing steps to analyze and extract impactful failures from network data sources.

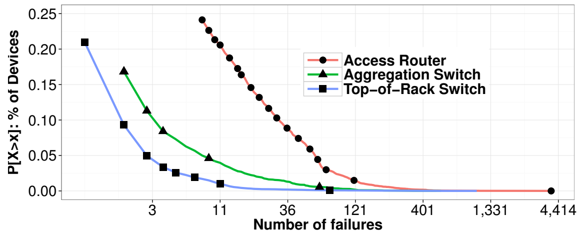


**Figure 4:** Complementary cumulative distribution (CCDF) plot of the number of failures logged by devices. The tail indicates devices that log thousands of failures.

3. Events being triggered by devices scheduled for replacement or those that have been detected as faulty by operators but awaiting repairs e.g., some devices logged more than 1000 device down notifications over few hours because the notification system did not suppress them during troubleshooting.

We build upon the methodology of Turner et al.[48] and our prior work[16] on utilizing low-level network events, but differentiate in four important ways. First, we apply a pipeline of event processing steps to analyze and correlate network event data sources (Figure 3). Second, we identify redundant events and analyze how they contribute towards measurement noise. Third, we remove events generated due to inactive links and planned maintenance to identify unexpected failures. Finally, we extract *impactful failures* by correlating network events with traffic loss, and infer their problem root causes from trouble tickets. Figure 3 shows the effectiveness of each processing step.

**Step 1**: The goal of the first step is to fix various timing inconsistencies. First, it groups all events with the same start and end time originating on the same interface with the same event description (thereby removing duplicate events). Next, by picking the earliest start and end times, multiple events within a 60 second time window on the same interface are grouped into a single event. This is done to avoid problems due to clock skews and log buffering. Finally, if two events originating on the same interface contain the same event description and have the same start time but different end times, they are grouped into a single event and assigned the earlier of the end times. We take the earliest times as events may not be marked as cleared long after their resolution.

**Step 2**: The second step filters all planned network changes based on a maintenance tracking system. Each network change is annotated with the time window, the device name and the type of maintenance being carried out. Network operators likely have a good understanding of problems being handled by scheduled maintenance and thus, we focus on analyzing device and link-level reliability due to unexpected outages.

**Step 3**: The third processing step removes redundant events due to devices that continue logging error messages when they are being troubleshooted or where the events were not suppressed even after the problem had been identified. Figure 4 shows the CCDF plot for different types of devices in our dataset. Observe that a small fraction of devices log up to thousands of failure events. To filter them, we apply the following technique based on discussion with operators: merge all events that have the same ticket identifier as events with the same ticket ID are likely to have the same symptoms.

**Step 4**: The final step aims to identify events causing service impact based on two rules: the event caused (i) *noticeable* application-level impact, or (ii) loss of traffic (i.e., a drop in the median traffic on the device/link during a failure compared to its median value in the recent past e.g., preceding 2-hour window). For the former, we leverage NetSieve [41] to extract the type

| Type | Mean (hrs) | Median (mins) | Q75 (hrs) | Q95 (hrs) | StdDev (days) |
|---|---|---|---|---|---|
| **AR** | 12.4 | 21.5 | 2 | 37.2 | 2.5 |
| **AGG** | 2.1 | 4.8 | 0.4 | 5.2 | 0.6 |
| **ToR** | 2.9 | 7.1 | 0.3 | 5.2 | 0.8 |

**Table 1:** Comparing TTR across ARs, AGGs and ToRs.

of application-level impact from trouble tickets. For the latter, we leverage the network traffic logs and performed hypothesis testing to validate that the median traffic value is robust to short-term traffic variations [40].

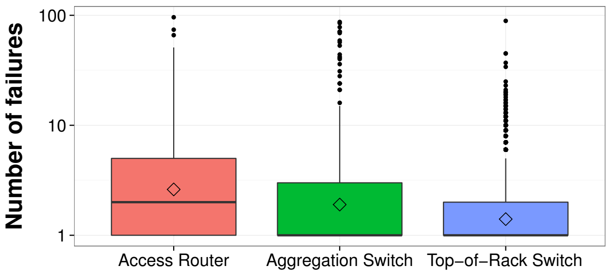
**Validation.** We performed ground truth validation to evaluate the fidelity of our failure analysis methodology. Specifically, we validate our methodology along two dimensions: (1) accuracy i.e., are all the processed events actionable? and (2) completeness i.e., did it miss any events from the ground truth data? For the former, we ensure that our result set includes *all* events deemed "actionable" by operators — we can recognize these actionable events by verifying if an operator attached a trouble ticket to it implying that the event was troubleshooted. For the latter, we leverage the high severity incident database described in §1. Because each such incident caused a service impact where the network redundancy was ineffective, we use this database as the ground truth. We compared our result set against the high severity incidents, and verified that (1) none of the events from this incident list were missed (i.e., no false negatives) and (2) in each case, network redundancy was in fact, unsuccessful in masking the failure.

# 4 Network Stamp Availability

To provide uptime service-level-agreement (SLA) of a geo-distributed service, a key requirement is to analyze the availability of a network stamp hosting the service and then compute the number of network stamps needed to meet the service SLA. In particular, we need to examine two key aspects: (a) the failure characteristics of the individual components comprising a network stamp, and (b) the effectiveness of network redundancy, typically deployed as a resiliency mechanism, in handling network failures. We then leverage these analyses to compute the network stamp availability.

## 4.1 Failure Characteristics of Building Blocks of a Network Stamp

We first analyze the number of failures per device across three types of network elements: Access routers, Aggregation switches and Top-of-Rack switches. Figure 5 shows a box-plot of the number of failures per device across the three device types. Notice that the mean number of failures per device is highest for ARs. ToRs exhibit the lowest mean number of failures due to their large population, which is consistent with our previous findings [16]. Though AGGs experience fewer failures per device with a median of one failure (table below Figure 5), the device population exhibits a high variability with a COV[1] of 2.4. This is also evident from the presence of outliers in Figure 5 (the dots present at the top of each box) indicating the presence of a "few bad" devices that log numerous failures.



| Type | Mean | Median | Q75 | Q95 | StdDev | COV |
|---|---|---|---|---|---|---|
| **AR** | 5.2 | 2 | 5 | 19.9 | 9.9 | 1.8 |
| **AGG** | 3.7 | 1 | 3 | 12.0 | 9.2 | 2.4 |
| **ToR** | 1.8 | 1 | 3 | 6.0 | 2.6 | 1.4 |

**Figure 5:** Comparing the number of failures per device per year across ARs, AGGs and ToRs. In the box-plot, the horizontal bolded line is the median; the box boundaries are interquartile ranges; the dots are outliers. The y-axis is on log scale.

We observe that most failures are short-lived which occur when the device unexpectedly reloads and then quickly comes back into an operational state. Consider the time-to-repair distribution for these device types shown in Table 1: both AGGs and ToRs have a small median time to repair of ≈5-7 minutes. We confirmed this observation from the trouble tickets associated with these events (see §5.1). ARs had the highest mean time to repair of ≈12 hours and a median of 21.5 minutes. This is a surprising result considering that ARs are positioned higher up in the network hierarchy and one expects that their problems get repaired relatively faster. We observed dominant problems related to network modules and switching fabric errors where the module either reloads unexpectedly or exhibits CRC packet errors. Besides line card failures, AGGs also exhibited soft-parity errors that cause the device to transition into an unexpected state. Soft-parity errors occur when the

[1] COV [8] shows the extent of variability to mean of the population and is defined as the ratio of the standard deviation $\sigma$ to the mean $\mu$. Distributions with COV $<$ 1 exhibit low-variance and vice versa.

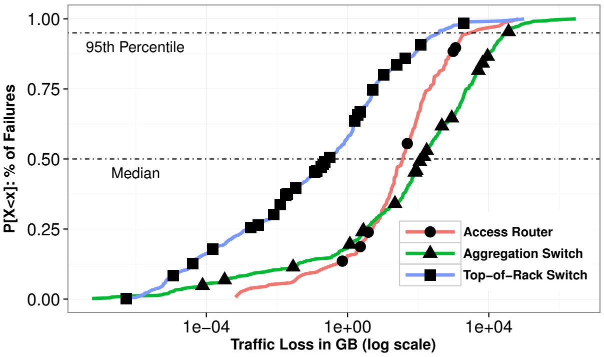


**Figure 6:** Distribution of the estimated traffic lost per day across ARs, AGGs, and ToRs. The x-axis is on log scale.

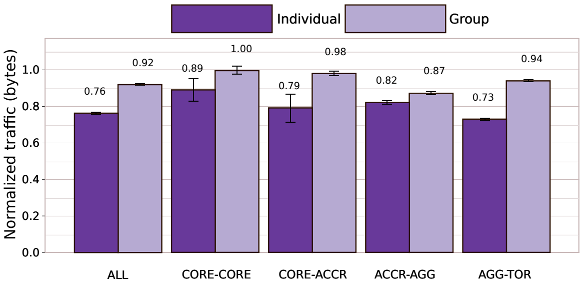


**Figure 7:** Normalized traffic (bytes) for failure events for individual devices and their redundancy groups

energy level within the chip changes, most often due to radiation (e.g., gamma rays, electro-magnetic interference from a neighboring device, electro-static discharge due to improper handling of the device). When referenced by the CPU, these errors cause the system to crash.

## 4.2 Effectiveness of Network Redundancy

In this section we analyze the effectiveness of redundancy at both intra- and inter-datacenter level in handling network failures. First, we estimate the traffic lost by impactful failure events and then compute the effectiveness of network redundancy at each layer of the datacenter hierarchy.

### 4.2.1 Estimating Traffic Loss

Quantifying the impact of a failure is difficult as it requires attributing discrete "outage" levels to annotations used by network operators such as *severe, mild*, or *some impact*. To circumvent this problem, we correlate the network event logs with link-level traffic measurements to estimate the impact of a failure event. However, it is still difficult to precisely quantify how much data was *actually* lost during a failure because of several complications: (i) traffic rerouted via alternate routes in datacenters, (ii) temporal variations in traffic patterns, and (iii) grey failures.

Building upon our prior work[16], we *estimate* the failure impact in terms of lost network traffic that would have been routed on a failed link in the absence of the failure. Specifically, we first compute the median number of bytes on links connected to the failed device in the time period preceding the failure, $Med_b$, and the median number of bytes during the failure, $Med_d$. Then the estimated median traffic loss per day for a failed device can be defined as:

$$\sum_{\forall\ events\ \in\ day} (Med_b - Med_d) \times failure\_duration \qquad (1)$$

where $failure\_duration$ denotes how long the device/link failure lasted.

Figure 6 shows the CDF of estimated traffic loss per day across ARs, AGGs and ToRs computed using Equation 1. We observe that the median number of bytes lost during failures is highest for AGGs (130 GB/day), but it is significantly less for ARs (38 GB/day) due to the "few bad apples effect" exhibited by AGGs as observed in §4.1. In comparison, the median traffic loss per day for ToRs is less than 1GB/day likely due to (a) relatively small bandwidth capacity compared to AGGs and ARs, (b) low downtime values (from §4.1), and (c) low traffic utilization at the ToR layer in the network hierarchy [16].

### 4.2.2 Analyzing Redundancy Effectiveness

We next analyze the effectiveness of network redundancy in mitigating the failure impact as it is a *defacto* resiliency mechanism to handle faults in datacenters. Within a redundancy group, one device is typically designated as active (primary) and the rest as standbys (backups). Other configurations are possible such as active-active pairs where both the devices carry traffic simultaneously. We observed several large redundancy groups comprising up to tens of Aggregation switches connected to a pair of Access routers in the network datacenter hierarchy.

To estimate the effectiveness of network redundancy, we first compute the ratio of median traffic (bytes) entering a device across all links during a failure and the median traffic entering the device before the failure, and then compute this ratio across all devices in the redundancy group where the failure occurred. Network redundancy is considered 100% effective if this ratio is close to one across a redundancy group. We refer to this ratio as *normalized traffic*.

Figure 7 shows the normalized traffic (bytes) at each layer of the datacenter network hierarchy. Overall, the median traffic carried at the redundancy group level is 92% compared with 76% at the individual level, an improvement of 21% in the overall median traffic as a result of network redundancy. While this redundancy is more effective at the AGG-ToR level (an improvement of 28.7%) and CORE-ACCR level (an improvement of 24%), it is relatively less effective at the ACCR-AGG level (an improvement of 6.09%). The maximum gain is observed at the CORE-CORE (inter-datacenter) layer where failures are completely masked due to redundancy. One reason is that network layers close to the root carry significant service traffic and connect application servers inside datacenters to users across the Internet. Hence, these layers are monitored closely by network operators to enable fast failure detection and troubleshooting in order to minimize service downtime.

**Computing number of network stamps to meet a service uptime SLA.** After computing the overall redundancy effectiveness $r$, we can leverage it to determine the minimum number $n_{min}$ of independent network stamps needed to meet a desired SLA uptime of a service. In particular, we solve for the minimum integer $n$ that satisfies the following equation:

$$1-(1-r)^n \geq SLA_{desired}$$
$$\Rightarrow log(1-SLA_{desired}) \geq nlog(1-r)$$

As $log(1-r) < 0 \, \forall \, r \in (0,1)$, dividing by $log(1-r)$ on both sides we get:

$$n \geq \frac{log(1-SLA_{desired})}{log(1-r)} \tag{2}$$

$$\Rightarrow n_{min} = \left\lceil \frac{log(1-SLA_{desired})}{log(1-r)} \right\rceil \tag{3}$$

where $\lceil \rceil$ is the ceiling function. For instance, to provide 99.9% availability (maximum 8.76 hours of downtime per year), solving the above equation yields $n = 3$. Similarly, for 99.99% availability (maximum 52 minutes of downtime per year) we get $n = 4$. Note that as expected, setting $n$ to higher values would likely yield diminishing returns in improving service availability.

**Findings (1)**: (1) The median number of bytes lost during failures is about 130 GB/day for ARs and 38 GB/day for AGGs while it is about 1 GB/day for ToRs. (2) Overall, the median traffic carried at the redundancy group level is 92% compared with 76% at the individual level, an improvement of 21% due to network redundancy. This redundancy is least effective at the AR-AGG level and is most effective at the Inter-datacenter level. (3) The number of independent network stamps for a desired uptime SLA of 3 9's (maximum 8.76 hours of downtime per year) is three and for 4 9's (maximum 52 minutes of downtime per year) is four.

# 5 Causes of Network Failures

In this section we analyze the root causes of intra- and inter-datacenter network failures.

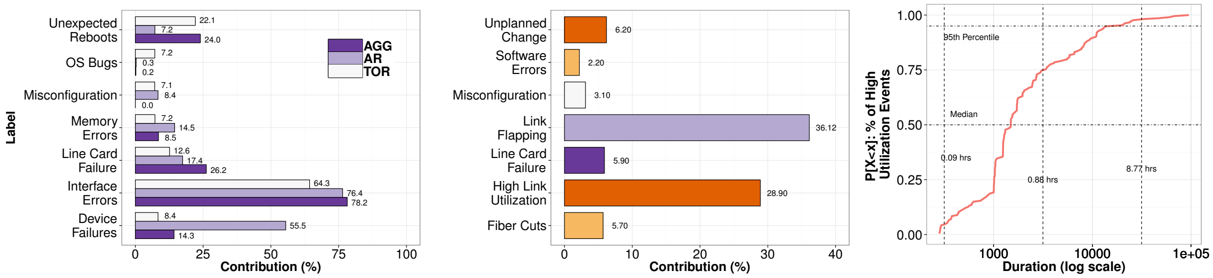
## 5.1 Intra-Datacenter

To determine the failure root causes, we leverage the information recorded by operators in trouble tickets attached to the network events. Specifically, we leverage NetSieve [41], an automated problem inference system that analyzes the free-form text in a trouble ticket to generate its synopsis: (1) the problems observed e.g., link down, misconfigured rule, switch in 'freeze' state, (2) the troubleshooting performed e.g., check cable, track configuration changes, verify BGP routes, and (3) the actions taken for resolution e.g., replaced the supervisor engine, reboot the device, clean the fiber.

Figure 8(a) shows the histogram of the top-k problems observed from trouble tickets associated with intra-datacenter failures. Observe that there is a broad range of problems such as hardware faults (e.g., device failures, memory errors), OS bugs, and misconfigurations (e.g., ARP conflict). Interface-level errors, network card problems, and unexpected reloads are prominent amongst all the three device types. Interface errors usually last for about 5-7 minutes as observed in Table 1. During these periods, we observe that the service would be available, but its users may experience high latency or packet drops. For instance, due to interface errors TCP may likely timeout and re-transmit in the slow-start phase thus degrading the service performance. Our discussion with operators revealed there are multiple reasons for interface errors such as faulty cable installation, faulty optical transceivers, and protocol convergence delays. In addition to interface errors and hardware problems, ToR failures were also due to OS-related problems and misconfigurations. For instance, in some cases a specially crafted IPv6 packet (e.g., Type-0 Routing Header[11] packets for source routing) was found to crash the device. In others, certain types of IPv4 packets (e.g., ICMP echo-requests) destined to a physical or virtual interface on the device caused a memory leak.

## 5.2 Inter-Datacenter

We analyze the network trouble tickets associated with Inter-DC link failures and found that link flapping (e.g., due to BGP, OSPF protocol issues and convergence) dominate the problem root causes (36%) as seen in Figure 8(b). Due to optical protection configured in some areas of the network, a physical layer problem might end up triggering an optical re-route — a technique to reduce the bandwidth loss by shifting existing lightpaths



(a) (b) (c)

**Figure 8:** (a) Intra-DC root cause analysis: Interface errors dominate across all device types. (b) Inter-DC root cause analysis: Link flapping and high link utilization are the major problems. (c) Distribution of the duration of the high link utilization events.

to new wavelengths, but without changing their route. However, it incurs a control overhead (of about 10 seconds), and more importantly, it risks a service disruption in the rerouted lightpaths [32]. Depending on the protocol timers, such an event is observed as a "link flap", yet the true underlying cause could be an optical re-route, possibly in response to a fiber cut (e.g., due to construction, bullet hole, vandalism, shark attack on under-sea cables). Therefore, it may not be possible in some cases to attribute the exact root cause. The second major root cause is high link utilization (29%). However, note that high utilization does not necessarily imply a physical circuit failure or take-downs, but it may be an indicator of packet errors; we analyze them in detail next. Software errors, misconfigurations, and unnotified maintenance were observed, but they did not constitute a significant fraction of root causes.

Next, we analyze the properties of links with high utilization. By examining trouble tickets associated with high utilization links, we observe that: (1) the percentage of error packets exceed the error threshold specified by operators, and (2) the traffic utilization of the link is above the specified utilization threshold. When there is insufficient capacity during a congestion period, LSP[2] re-routing occurs during which the traffic is switched to an alternate LSP set up after the failure. The new route is selected at the LSP head-end (router that requested the LSP establishment) and may reuse intermediate nodes in the original route. This alternate route maybe computed either on demand or pre-computed, and stored for use when a failure is reported. There is, however, a risk that the alternate route may become out of date due to other changes in the network. This can be mitigated to some extent by periodic recalculation of idle alternate routes. In practice, about 10-25 minutes are allowed for LSP re-routing. However, we observe in Figure 8(c) that the average incident duration is about 1.27 hours while the median is 0.41 hours with the 95P value >3 hours, which is about 2x-3x higher than the expected value. Further, 80% of the high utilization events took more than 15 minutes to resolve and 50% of the events took more than half an hour.

Longer downtimes can be attributed to scenarios where LSP re-routing had no alternate routes to compute on demand which caused higher switching times. Long-lived congestion is expected when there is a surge in traffic demand that is relatively long-lived, and the bottleneck is not the transit capacity but the capacity at a source or sink for the traffic. This is believed to happen when there is a significant outage with lack of sufficient redundant capacity. Although it was not possible to pin point the exact root cause behind the longer downtimes in our dataset, one potential cause from discussion with operators was attributed to congestion. Rather than the entire circuit going down under congestion, the links logged errors continuously (a significant fraction of packets were being discarded) for long periods of time which inflated the event duration. Note that these events logged as "failure events" do not imply physical circuit breaks but they risk performance degradation due to congestion. The packet discards indicate that during the congestion period, LSP rerouting did not have an alternate path with sufficient capacity to forward the excess traffic. There are many reasons why congestion could arise:

- **Product-related**: During product migration (cloud service changing datacenters) or launch (new software being released), there is a high volume of traffic.
- **Traffic Shift**: Workloads arising due to unexpected service outages or bulk data transfer e.g., web crawl documents, periodic data backups.
- **Port-name Cleanup**: As part of improving network meta-data consistency and integrity, operators perform port-name cleanup on a regular basis. If these changes are not reflected higher up in the topology, then routes may not be available on demand.

[2] In MPLS wide-area networks, data transmission occurs on label-switched paths (LSPs). LSP is a path through the network from an ingress to an egress router established through the distribution of labels (using label distribution protocol (LDP) or piggybacked on routing protocols like BGP) that define hop-by-hop forwarding.

| Type | Pr[$1^{st}$] | Pr[$2^{nd}$—$1^{st}$] | Pr[$3^{rd}$— $2^{nd}$] | Eff? |
|---|---|---|---|---|
| AR | 1 in 4.0 | 1 in 5.2 | 1 in 5.8 | ✓ |
| AGG | 1 in 3.7 | 1 in 6.1 | 1 in 8.6 | ✓ |
| ToR | 1 in 17.1 | 1 in 7.7 | 1 in 5.4 | — |

**Table 2:** The (conditional) probability of device failures. The last column indicates if the repairs were effective.

**Findings (2)**: (1) Link flapping and interface errors dominate problem root causes across all device types. (2) Other dominant causes are hardware failures, unexpected reboots and misconfigurations. (3) Links with high utilization exhibit 2x-3x higher downtime than expected.

# 6 Failure Analysis and Modeling

In this section we aim to answer the following key questions: (1) Are repairs effective in fixing problems? (2) Are failures transient or independent? If not, what are their properties? (3) How soon does a device fail after experiencing a failure? and (4) Does the length of fiber links affect the probability of link failure?

## 6.1 Are repairs effective?

We begin by quantifying the probability that a device will fail multiple times in its lifetime based on the observed failure rates. For each device platform, we analyze the conditional probability that a device will fail if it previously experienced one or more failures. Table 2 shows the conditional probabilities of successive device failures split by the three device types. An increase in the probability with every subsequent failure indicates that actions taken to "fix" failures are not effective and vice versa. We make the following observations from Table 2:

- **Access Routers**: ARs in general exhibit a decreasing trend to fail indicating that the repairs at each failure level are effective. These devices exhibited a 1 in 4 chance of a first failure during the observation period. After a device has failed once, its failure probability decreases by a factor of ≈1.5, and the probability continues to decrease with subsequent failures indicating that it is favorable to consider repairing devices of this type when they fail. However, note that for some device generations, repairs may not be as effective e.g., the probability of failure likely increases with each subsequent repair for old devices close to their end-of-life as expected.
- **Aggregation Switches**: AGGs exhibit a decreasing failure trend similar to ARs. However, some old generations of AGGs showed an increased probability of successive failures.

| Device Type | KS-Value | p-value |
|---|---|---|
| Access Routers | 0.4952 | $2.2x10^{-16}$ |
| Aggregation Switches | 0.665 | $2.2x10^{-16}$ |
| Top-of-Rack Switches | 0.7236 | $2.2x10^{-16}$ |

**Table 3:** KS Test to determine whether failure inter-occurrence times are exponentially distributed.

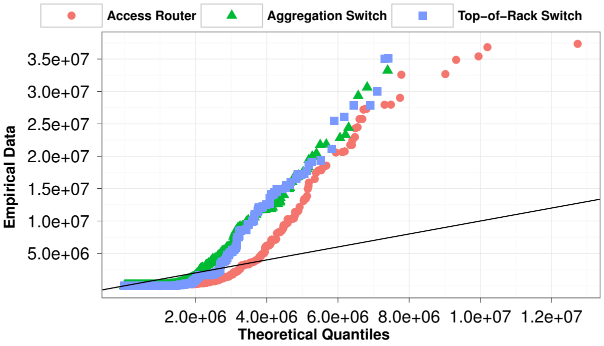


**Figure 9:** Q-Q plot of inter-arrival times of device failures of ARs, AGGs and ToRs in a datacenter — the plot indicates that they are not exponentially distributed.

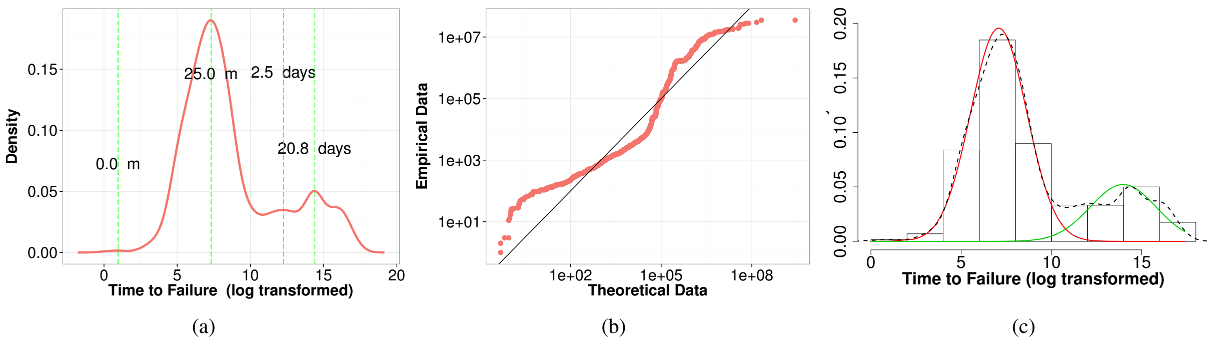
- **Top-of-Rack Switches**: The most surprising result relates to ToRs — they exhibit the worst behavior of increase in failure probability after every subsequent failure. This increase indicates that repairs carried out as a response to failures are not quite effective in mitigating the failure root cause. One likely reason is that ToRs have a low priority for repair and thus, a quick-fix solution (e.g., reboot) typically applied may delay finding the true root cause of the problem.

**Findings (3)**: (1) Repairs were relatively more effective for ARs and AGGs. (2) ToRs exhibit an increase in probability of device failure after repair indicating that their repairs are not quite effective.

## 6.2 How to model failures?

We next answer the question if failure occurrences are independent or memoryless[3]. We use a Q-Q plot [49] to check if the empirical occurrence rates (distribution shown in Table 4) come from an exponential distribution. If the data follow an approximately straight line with slope 1 and intercept 0, the observed values are said to be drawn from the exponential distribution. However, our data does not follow this trend as Figure 9 shows. Finally, we perform the KS-test [28] on the failure inter-arrival times of the three device types; Table 3 shows the test results. We observe that the *null hypothesis* that failure inter-arrival times are exponentially distributed can be rejected at a significance level of 0.05.

[3] Memoryless property indicates that failure inter-arrival times are exponentially distributed



(a) (b) (c)

**Figure 10:** Modeling Time to Failure for ToRs. (a) Kernel density plot of the log power transformed TTF, (b) Fit of a single log-nornal distribution, (c) Fit of a mixture of two log-normal distributions.

| Type | Mean (days) | Median (hrs) | Q75 (days) | Q95 (month | StdDev (months) |
|---|---|---|---|---|---|
| **AR** | 19.3 | 14.5 | 6.1 | 4.0 | 1.9 |
| **AGG** | 11.2 | 0.2 | 1.5 | 2.2 | 1.2 |
| **ToR** | 11.1 | 0.6 | 0.3 | 2.9 | 1.2 |

**Table 4:** Comparing TTF across ARs, AGGs and ToRs.

**Failure Modeling.** As Table 4 indicates, the range of TTF durations can be several orders of magnitude across devices. We build upon our prior work [40] of using the Box-Cox transformation [42] to model the time to failure for ToRs as an illustration.

Figure 10(a) shows the kernel density plot of the log-power transformed TTF. We observe that the distributions are right skewed and not unimodal. We find that the main peaks occur at 25 minutes for TTF with relatively smaller secondary peaks occurring at about 2.5 days and 20 days. These findings indicate that there are two or three qualitatively different *types* of failures in effect, respectively. The peak at 25 minutes indicates that short-term failures such as connection errors, interface flaps, and unexpected reboots as seen in Figure 8(a) dominate, and the secondary peak at 20 days indicates problems due to both hardware faults (e.g., line card, device failures) and software bugs.

Similar to our observation for load balancers [40], existing heavy-tailed distributions did not fit our ToR failure data (the empirical data significantly deviated from the '$y = x$' line in the Q-Q plot in Figure 10(b)). Therefore, we leverage a two-component mixture model to approximate the failure data. Assume that the real-valued variables $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are a simple random sample of time periods from a finite mixture of $m > 1$ arbitrary distribution components. The density of each $\mathbf{X}_i$ can then be written as:

$$h_\theta(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{x}_i), \mathbf{x}_i \in \mathbf{R}^r \qquad (4)$$

| Type | $\lambda$ | $\mu_{ln_1}$ | $\sigma_{ln_1}$ | $\mu_{ln_2}$ | $\sigma_{ln_2}$ |
|---|---|---|---|---|---|
| **TOR** | 0.762147 | 7.080109 | 1.553494 | 13.973199 | 1.819118 |
| **AGG** | 0.184676 | 3.965197 | 0.165990 | 9.01190 | 4.125347 |
| **AR** | 0.333407 | 1.831425 | 1.553494 | 12.498570 | 2.395274 |

**Table 5:** Parameters for the two-component lognormal mixture distribution for different device types

where $\theta = (\lambda, \phi) = (\lambda_1, \ldots \lambda_m, \phi_1, \ldots, \phi_m)$ denotes the model parameter and $\sum_{j=1}^{m} \lambda_m = 1$. If we assume that $\phi_j$ are drawn from some family $\mathscr{F}$ of univariate log-normal density functions on $\mathbf{R}$ given by $\mathscr{F} = \{\phi(\cdot|\mu_{ln}, \sigma_{ln}^2\}$, where $\mu_{ln}$ and $\sigma_{ln}$ denote the mean and standard deviation in log scale, then the model parameter reduces to $\theta = (\lambda, (\mu_{ln_1}, \sigma_{ln_1}^2), \ldots, (\mu_{ln_m}, \sigma_{ln_m}^2))$. By substituting these parameters, Equation (4) can be written as:

$$h_\theta(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \frac{1}{\sigma_{ln_j} \mathbf{x}_i \sqrt{2\pi}} e^{-\frac{(ln(\mathbf{x}_i) - \mu_{ln_j})^2}{2\sigma_{ln_j}^2}}, \mathbf{x}_i \in \mathbf{R}^r \qquad (5)$$

For a two-component lognormal mixture, Equation (4) becomes: $\lambda f(\mu_{ln_1}, \sigma_{ln_1}) + (1 - \lambda) f(\mu_{ln_2}, \sigma_{ln_2})$. Subsequently, we use Expectation-Maximization (EM) [33] to obtain the model parameter $\lambda$. Table 5 gives the values of the parameters for the two-component lognormal mixture distribution to fit the failure data for the three device types. Figure 10(c) shows how this model fits our ToR failure data (at a log-likelihood of -3389.3); the dotted-line is the kernel density curve of our data and the solid lines are the individual mixture components. We observe that our model provides a good approximation of the real-world data of ToR, AGG and AR failures in cloud datacenters.

**Findings (4)**: (1) Device failures are *not* memoryless i.e., they are not independent. (2) AGGs exhibit the "few bad apples" effect. (3) The TTF kernel density of ToRs shows the highest peak at 25 minutes indicating dominance of short-lived problems such as connection errors, unexpected

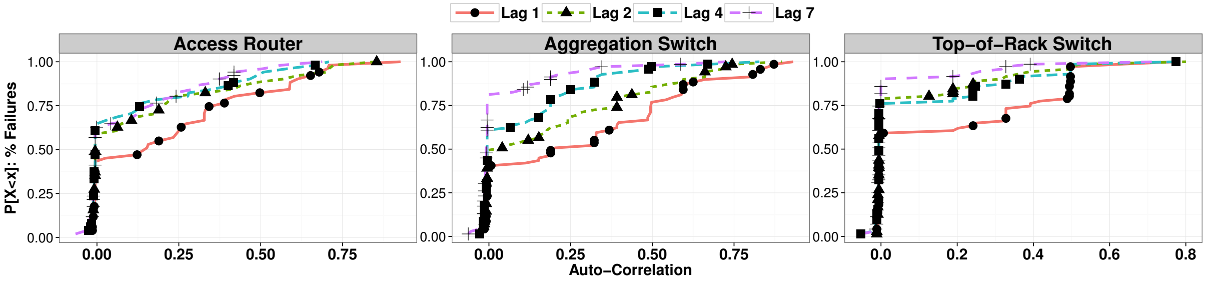


**Figure 11:** Correlations of device failures over a week period for different device types at different time lags (curves towards the top-left show low correlation).

reloads and interface flaps. (4) A univariate lognormal distribution is unsuitable (poor-fit) to model TTF of network device failures; a two-component lognormal mixture distribution provides a good approximation.

## 6.3 Are failures bursty?

To understand how quickly devices fail after experiencing a failure, we compute the auto-correlation function for the number of failures observed per device on a daily-basis. For each device, we construct a binary time series as tagging 1 on a day if it exhibited a failure on that day and 0 if the device was functioning normally.

Figure 11 shows the CDF of the auto-correlation values for different device types at different lag levels (shift in the time series). ToRs exhibit a short term stable behavior i.e. they do not exhibit any statistically significant correlation with respect to next day failures indicating that fixes deployed are at least temporarily effective. However, over the long term, as described in the previous section, their long-term reliability trends show an increased probability of failure.

ARs and AGGs indicate that for the devices that do fail multiple times (graph omitted), 20%-30% of this population is likely to fail the next day or within a week of getting fixed. We observed that this happens when either the deployed fix is ineffective (e.g., a "reboot" was performed as a quick-fix) or when the root cause was mis-diagnosed (e.g., the supervisor engine was faulty, but the cable was replaced).

**Findings (5)**: All device types exhibit some amount of "burstiness" in their failure patterns with ToRs showing the least. After one failure, probability of subsequent failures is higher in the near time window. However, the probability of multiple failures is quite low ($< 0.05$) and when devices fail multiple times, they do so within one week of getting fixed likely due to ineffective repairs and problem mis-diagnosis.

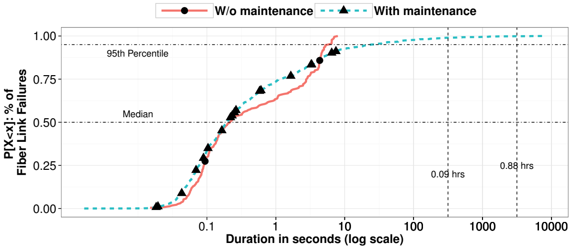


**Figure 12:** Link out per km of long-haul links

| Attribute Pair | Pearson Correlation |
|---|---|
| Distance vs. Segments | 0.854 |
| #Failures vs. Distance | 0.238 |
| #Failures vs. Segments | 0.285 |

**Table 6:** Pearson correlation

## 6.4 Is fiber length correlated to failures?

Intuitively, longer the link, higher is the probability of at least one of its components (e.g., segments) to fail. We use the Pearson Product-Moment [5] to analyze if there is a correlation between the number of failures observed on a fiber and #segments per circuit;. Table 6 shows the results. Surprisingly, we find that the fiber length has no statistically significant correlation with the number of failures observed. We attribute this result to the fact that these components exhibit high reliability [18] and fail independently (likely due to issues such as construction, rodent bites and under-water fiber cuts). Hence, the overall reliability even for very-long links is not affected.

To understand how many seconds a link is down per kilometer, we define *link out per km* as the ratio between the duration that a link is down to the length (in kms) of the fiber involved in the failure event. Figure 12 shows the distribution of this metric for *all* fiber link failures and the ones not due to maintenance. We observe a median link out time of 0.4 seconds/km and a 95P value of 5.9 seconds/km for the latter and ≈26.7 seconds/km for

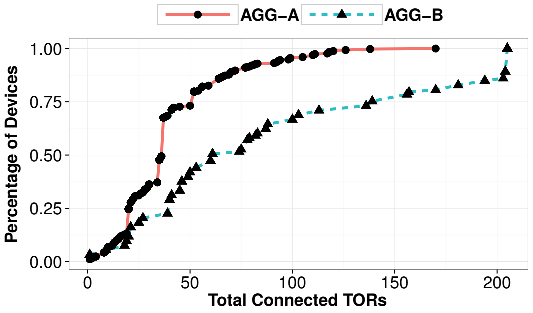


**Figure 13:** Distribution of the number of ToRs connected to two AGG platforms.

the former with a long tail for maintenance events. Computing the Pearson correlation between fiber length and link out per km yielded -0.13 for all events with maintenance and -0.16 for impactful events indicating that they are not statistically correlated.

**Findings (6)**: Fiber length has no statistically significant correlation with the number of failures observed.

# 7 Capacity vs. Availability

In network design, a conventional approach is to adopt a multi-layer architecture that breaks up the network into small, more manageable Layer-2 domains and then use a spanning tree to provide redundancy and network load sharing. The size of these domains is kept small to reduce the delays in spanning-tree convergence (which is sensitive to the network diameter). Due to the Layer-2 topology, the network can scale at this layer simply by adding more switches. However, the drawback is that scaling Layer-2 domains increases the *fault domain size* e.g., a broadcast storm caused by a malfunctioning device or human error can cause failure of the entire subtree. Further, to avoid loops, all links cannot be in a forwarding state at all times because broadcast packets risk saturating the VLAN, thereby adversely affecting the network performance.

This raises a fundamental question of *scale-up* vs. *scale-out* for operators to deliver services in a cost-effective manner: *Do we deploy high-density, expensive Aggregation switches that can provide connectivity to hundreds of ToRs or leverage small port-count, low-cost commodity switches (perhaps with lower reliability), but deploy them in large numbers?* Specifically, we aim to analyze how does the availability of Layer-2 AGGs depends on the number of ToRs connected to it.

Figure 13 shows the distribution of the number of ToRs connected to two platforms of AGGs, AGG-A and AGG-B, in our dataset. Observe that a small percentage (5%-25%) of devices serve more than 100 ToRs. To understand the tradeoff between capacity and availability, we divide each platform of Aggregation switches into four categories based on quartiles on the ToR count (i.e., the first quartile forms the first category and so on). In each category, we compute the availability of the population based on the number of logged failure events.

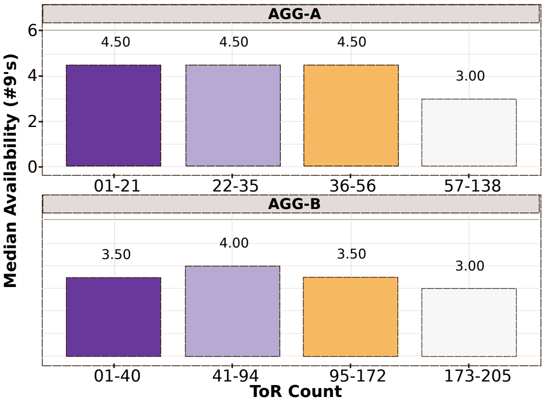


**Figure 14:** Median availability based on the number of ToRs connected to the two AGG platforms.

Figure 14 shows the median availability of AGGs based on the number of ToRs they are connected to. Notice that the category-1 devices for AGG-A exhibit 4.5 9's availability. By analyzing the trouble tickets of AGG-A devices in this category, we observed that the time to troubleshoot their problems was the lowest likely due to small number of ToRs resulting in high availability. For AGG-B, the availability in this category was relatively less of 3.5 9's due to higher time to debug platform-specific problems and relatively a wider range of up to 40 connected ToRs compared to AGG-A. f For both platforms, category-2 exhibited about the same availability as category-1 with a slight increase for AGG-B value of 4 9's. We also observed similar results for category-3 where AGG-A exhibited 4.5 9's availability while AGG-B exhibited 3.5 9's availability. We observed from the tickets of AGG-B devices in this category (having a ToR count range of 95-172) that the AGGs were being provisioned with new ToRs which increased the likelihood of a failure during troubleshooting e.g., due to operator mistakes.

In category-4, we observe the lowest availability of 3 9's across both platforms. Note that many of these devices were long standing in production whose ToR provisioning was close to port capacity. Using the tickets associated with this category, we found that maintenance becomes harder and more error-prone for high ToR counts as also observed for category-3 AGG-B devices. Further, due to these devices operating close to capacity, it was not feasible to find alternate available paths to re-route the entire traffic load in case of a failure. This resulted in additional downtime where the operators had to provision a temporary set of replacement devices and then configure them to route the service traffic.

**Findings (7)**: Layer-2 switches exhibit high availability when about half of their port capacity is utilized (in terms of ToR count). However, the availability significantly decreases as the ToR count gets close to the full switch capacity. Therefore, to deliver highly-reliable and cost-effective services, scale-out switches with low to medium port density may deliver higher availability in comparison to their expensive, higher capacity counterparts.

# 8 Research Implications

In this section we discuss the research implications based on our study to improve network reliability for geo-distributed services.

**DNS-redirection based network load balancing**: To mask service outages (§1) and to avoid overload at any datacenter site, one approach is to assign a URI to the cloud service whose DNS resolution maps to a set of globally distributed monitoring servers e.g., in a content distribution network. These servers dynamically re-route requests to the edge datacenter hosting the service that will provide the lowest latency based on a client's location. The monitoring servers track each datacenter hosting site via heartbeat signals and can further execute a specified set of mini-transactions [1] on the service to continuously check its performance and availability.
**Link Bundling and Wavelength Provisioning**: To reduce high utilization events (§5.2) on inter-DC links and to avoid the undesirable latency changes due to rerouting (e.g., caused by a fiber cut) at the optical layer, several techniques can be explored:
• Link Bundling: Link bundling, widely used in the context of MPLS [24] traffic engineering, combines multiple links into a single logical channel to increase aggregate bandwidth and fault tolerance. For fiber links, it can significantly reduce LSP fragmentation which arise from large flows being unable to fit on individual circuits. In particular, bundling allows creating many smaller, parallel LSPs between core routers, bringing the bandwidth to a packable number such as 2Gbps per LSP. The resulting large number of LSPs can then be re-routed occasionally and fit across a fragmented 10G optical mesh.
• Automatic Bandwidth: MPLS auto-bandwidth feature measures the traffic flows through the LSP, adjusting the bandwidth based on measured traffic and defined parameters on a per-LSP basis. While it allows the network to react faster to sudden traffic spikes without manual intervention, there are two key challenges of (a) managing significant routing churn by small-sized LSPs, and (b) tuning parameters to automatically create/delete LSPs.
• Mirror Physical Topology: Wavelength provisioning and link bundling can be planned so that bundles are created, mirroring the underlying physical topology. In this way, the reliability of an optical network is better modeled by the IP router topology. When there is an optical failure, the whole link bundle will go down, and should therefore be unavailable from a network and capacity planning point of view. This helps reduce complexity in understanding and mitigating failures because the IP topology closely mirrors the optical topology.
**Techniques to improve network redundancy**: Our analysis of redundancy effectiveness (§4.2) revealed the following problems to cause unsuccessful failovers:

- Misconfigurations and software version mismatch between primary and secondary redundant pairs.
- Faulty failovers when the backup exhibited a problem unrelated to primary failure (e.g. due to software errors, protocol bugs etc.)
- Faulty cables where the cable connected to the backup showed a high error rate on failover

Solving these broad range of problems requires exploring several research directions. Techniques from software engineering such as static analysis which found success in detecting BGP misconfigurations [14] can be explored for checking configurations of Layer-2 and Layer-3 network devices. Similarly, proactive fault injection [7] techniques can be leveraged to randomly shoot down network elements and testing service resilience in masking them.
**Repair vs. replace**: §6.3 indicated that when devices fail multiple times, they do so within one week of getting fixed. Therefore, finding and replacing the "*few bad apples*" will proactively help avoid serious problems. In addition, the analysis in §6.2 indicates that failures are not memoryless and while repairs were effective for ARs and AGGs, the probability of successive failure for ToRs increased. Thus, while a naive approach is to *immediately* replace a failed ToR, in practice this decision should be driven by two key factors: (1) Computing a Cost of Ownership (COO) [12] for devices to include their capital, operational, and repair and maintenance costs; §4.1 provides empirical results on some of these metrics, and (2) adopting a data-driven approach to compute the conditional probability $P_{N+1|N} = P((N+1)^{th} failure|N^{th} failure)$ for a device type/platform and then comparing it with both a threshold $\delta$ based on the network device platform's annualized failure rate and $P_{N|N-1}$. The intuition behind the latter is that if $P_{N+1|N} > \delta * P_{N|N-1}$, the probability of the device experiencing a subsequent failure is higher and thus it becomes a candidate for replacement.

## 9 Related Work

This paper is one of the first large-scale study of failures of network stamps from a service perspective, and the first study characterizing failures on inter-datacenter links in a cloud service provider. Several of our analyses have not been previously examined such as computing the number of independent network domains to meet uptime SLAs, checking whether network failures are memoryless or bursty, analyzing how does Aggregation switch port capacity impact their reliability, finding correlation of fiber length with failure rate, and studying high-utilization of inter-DC links.

**Datacenter Failures**: Failures in datacenters, in general, have received significant attention in the recent years [21, 25, 30, 37, 45, 48]. Our event processing methodology in §3.2 draws some similarity with the analysis carried out by Turner et al. [48] and Gill et al. [16]. Turner et al. [48] used router configurations, syslog and email records to analyze network failures in the CENIC network. Gill et al. [16] study intra-datacenter network failures in a large cloud provider, but they do not analyze reliability of network stamps from a service viewpoint or characterize failures of inter-datacenter links. Sherry et al. [46] conduct an operator survey of middlebox failures across several enterprise networks.

**Inter-Datacenter Links**: Much of the earlier work in analyzing inter-datacenter links focused on characterizing traffic flows [10, 26], providing bandwidth-on-demand [29] or optimizing traffic flows [27, 29]. Laoutaris et al. [26] present a system for bulk data transfer over wide area that employs a network of storage nodes and uses a store-and-forward algorithm to schedule data transfers. Chen et al. [10] characterize inter-datacenter traffic using anonymized NetFlow datasets collected at the border routers of Yahoo! data centers. They find that peak traffic volumes between datacenters are dominated by non-interactive, bulk data transfers.

Mahimkar et al. [29] present a globally reconfigurable photonic network between datacenters that improves operational flexibility by providing a bandwidth-on-demand service. Li et al. [27] present a scheduling scheme that considers both bandwidth utilization and ISP friendliness to reduce the inter-domain traffic. Unlike most work in this area, we focus primarily on analyzing long-haul link failures and studying properties of inter-datacenter links with high utilization. Perhaps, the closest work to ours is by Kandula et al. [22], who analyze when and where congestion happens inside datacenters but they do not consider inter-datacenter traffic. In an extended abstract [39], we performed a preliminary analysis of the causes behind intra- and inter-datacenter network failures.

**Hardware Failures**: There have been several prior studies on hardware failures (e.g., [4, 19, 35, 38, 43, 44] and the references therein), but they do not consider network failures in datacenters. Our findings on conditional failure probability relate to the findings in recent studies on DRAM errors [44] and disk-subsystem failures [19]. By contrast, we found diverse trends across different device types in our dataset which could be leveraged in a data-driven approach to decide whether to repair or replace a device. Nightingale et al. [35] study desktop failures running Windows and find that PC failures are not memoryless, consistent with our observation for the three types of network devices.

## 10 Conclusion

This paper presents one of the first large-scale study of cloud network failures at both intra-datacenter and inter-datacenter layers. We find that network failures cause significant impact to cloud services, dominated by connectivity loss problems and service errors. The main takeaways from our study are: (1) A service hosted on a single, large network stamp may risk low availability because network redundancy is least effective at the AR-AGG layer. To improve availability, build a small number of independent network stamps — three for 99.9% availability and four for 99.99% availability; (2) Network redundancy is most effective at the Inter-datacenter level. However, long-haul links exhibit 2x-3x higher downtime than expected under high utilization; (3) Scale-out switches with low to medium port density may deliver relatively higher availability in comparison to their expensive, higher capacity counterparts; (4) Network device failures are not memoryless and exhibit the "few bad apples" effect; techniques such as regression analysis and trend analysis can be leveraged to identify and troubleshoot the most failure-prone devices; (5) Interface errors, hardware failures and unexpected reboots dominate the problem root causes; and (6) Top-of-Rack switches exhibit an increase in probability of a successive device failure after repair motivating the need to develop automated correlation and diagnosis techniques. We hope that our work sheds light on answering several key questions to improve network reliability for geo-distributed services.

## 11 Acknowledgments

We thank our shepherd Matthew Caesar and the anonymous reviewers for their feedback. We are grateful to Vijay Gill and Darren Loher for their insightful comments. This work was done in part during Rahul's internship at Microsoft Research, Redmond.

# References

[1] Keynote Web Performance Testing. http://goo.gl/khl9Q.

[2] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan. Volley: Automated Data Placement for Geo-distributed Cloud Services. In *Proceedings of NSDI*. USENIX Association, 2010.

[3] Amazon. Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region. http://goo.gl/yUlTJ, May 2011.

[4] L. Bairavasundaram, A. Arpaci-Dusseau, R. Arpaci-Dusseau, G. Goodson, and B. Schroeder. An Analysis of Data Corruption in the Storage Stack. *Proceedings of ACM Transactions on Storage (TOS)*, 4(3), 2008.

[5] G. Box, J. Hunter, and W. Hunter. *Statistics for Experimenters: Design, Innovation, and Discovery*. Wiley, 2005.

[6] E. A. Brewer. Lessons from Giant-Scale Services. *Internet Computing, IEEE*, 5(4):46–55, 2001.

[7] J. Brodkin. Netflix attacks own network with "Chaos Monkey" - And now you can too. http://goo.gl/XhiKM, July 2012.

[8] C. E. Brown. Coefficient of Variation. In *Applied Multivariate Statistics in Geohydrology and Related Sciences*, pages 155–157. Springer, 1998.

[9] J. Case, M. Fedor, M. Schoffstall, and J. Davin. Simple Network Management Protocol. http://goo.gl/az3Fv, May 1990.

[10] Y. Chen, S. Jain, V. Adhikari, Z. Zhang, and K. Xu. A First Look at Inter-data Center Traffic Characteristics via Yahoo! Datasets. In *Proceedings of INFOCOM*. IEEE, 2011.

[11] S. Deering and R. Hinden. Internet Protocol, Version (IPv6) Specification. RFC 2460.

[12] L. Ellram. Total Cost of Ownership: An Analysis Approach for Purchasing. *Journal of PDLM*, 1995.

[13] D. Etherington. Dropbox Currently Experiencing Widespread Service Outage. http://goo.gl/rszmb, May 2013.

[14] N. Feamster and H. Balakrishnan. Detecting BGP Configuration Faults with Static Analysis. In *Proceedings of USENIX NSDI*, 2005.

[15] S. G. and I. B. Websites Scramble as Hurricane Sandy Floods Data Centers. http://goo.gl/zOXDb, October 31 2012.

[16] P. Gill, N. Jain, and N. Nagappan. Understanding Network Failures in Datacenters: Measurement, Analysis, and Implications. In *Proceedings of SIGCOMM*, 2011.

[17] A. Greenberg, J. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta. VL2: A Scalable and Flexible Datacenter Network. *ACM SIGCOMM CCR*, 2009.

[18] H. Jiang, F. Kéfélian, S. Crane, O. Lopez, M. Lours, J. Millo, D. Holleville, P. Lemonde, C. Chardonnet, A. Amy-Klein, et al. Long-distance Frequency Transfer Over an Urban Fiber Link Using Optical Phase Stabilization. *JOSA B*, 25(12), 2008.

[19] W. Jiang, C. Hu, Y. Zhou, and A. Kanevsky. Are disks the dominant contributor for storage failures?: A Comprehensive Study of Storage Subsystem Failure Characteristics. *TOS*, 2008.

[20] D. Johnson. NOC Internal Integrated Trouble Ticket System. http://goo.gl/eMZxX, January 1992.

[21] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl. Detailed Diagnosis in Enterprise Networks. In *ACM SIGCOMM CCR*, 2009.

[22] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken. The Nature of Data center Traffic: Measurements & Analysis. In *Proceedings of SIGCOMM*. ACM, 2009.

[23] D. C. Knowledge. Data Center Global Expansion Trend. http://goo.gl/SOvtA, November 2012.

[24] K. Kompella, L. Berger, and Y. Rekhter. Link Bundling in MPLS Traffic Engineering (TE). 2005.

[25] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental Study of Internet Stability and Backbone Failures. In *Proceedings of IEEE Fault-Tolerant Computing*, 1999.

[26] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez. Inter-datacenter Bulk Transfers with NetStitcher. In *Proceedings of SIGCOMM*, 2011.

[27] Y. Li, H. Wang, P. Zhang, J. Dong, and S. Cheng. D4D: Inter-datacenter Bulk Transfers with ISP Friendliness. In *IEEE CLUSTER*, 2012.

[28] H. Lilliefors. On the Kolmogorov-Smirnov Test for the Exponential Distribution with Mean Unknown. *Journal of the American Statistical Association*, 64(325), 1969.

[29] A. Mahimkar, A. Chiu, R. Doverspike, M. Feuer, P. Magill, E. Mavrogiorgis, J. Pastor, S. Woodward, and J. Yates. Bandwidth On Demand for Inter-Data center Communication. In *HotNets*. ACM, 2011.

[30] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, Y. Ganjali, and C. Diot. Characterization of Failures in an Operational IP Backbone Network. *IEEE/ACM TON*, 2008.

[31] M. McCloghrie, K. ad Rose. Management Information Base for Network Management of TCP/IP-based internets. RFC 1213.

[32] G. Mohan and C. Murthy. Lightpath Restoration in WDM Optical Networks. *Network, IEEE*, 14(6), 2000.

[33] T. K. Moon. The Expectation-Maximization Algorithm. *Signal Processing Magazine, IEEE*, 13(6):47–60, 1996.

[34] J. Mudigonda, P. Yalagandula, J. Mogul, B. Stiekes, and Y. Pouffary. NetLord: A Scalable Multi-tenant Network Architecture for Virtualized Datacenters. In *Proceedings of ACM SIGCOMM*, 2011.

[35] E. Nightingale, J. Douceur, and V. Orgovan. Cycles, Cells and Platters: An Empirical Analysis of Hardware Failures on a Million Consumer PCs. In *Proceedings of the Sixth Conference on Computer Systems*. ACM, 2011.

[36] R. Niranjan Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat. PortLand: A Scalable Fault-Tolerant Layer-2 Data center Network Fabric. In *SIGCOMM CCR*. ACM, 2009.

[37] V. Padmanabhan, S. Ramabhadran, S. Agarwal, and J. Padhye. A Study of End-to-End Web Access Failures. In *Proceedings of ACM CoNEXT*, 2006.

[38] E. Pinheiro, W. Weber, and L. Barroso. Failure Trends in a Large Disk Drive Population. In *Proceedings of FAST*, 2007.

[39] R. Potharaju and N. Jain. An Empirical Analysis of Intra-and Inter-datacenter Network Failures for Geo-distributed Services. In *Extended Abstract Proceedings of ACM SIGMETRICS*. ACM, 2013.

[40] R. Potharaju and N. Jain. Demystifying the Dark Side of the Middle: A Field Study of Middlebox Failures in Datacenters. In *Proceedings of the 13th ACM SIGCOMM Conference on Internet Measurement*, 2013.

[41] R. Potharaju, N. Jain, and C. Nita-Rotaru. Juggling the Jigsaw: Towards Automated Problem Inference from Network Trouble Tickets. In *Proceedings of USENIX NSDI*, 2013.

[42] R. Sakia. The Box-Cox Transformation Technique: A Review. *The Statistician*, pages 169–178, 1992.

[43] B. Schroeder and G. Gibson. Disk Failures in the Real World: What does an MTTF of 1,000,000 hours mean to you. In *Proceedings of FAST*, 2007.

[44] B. Schroeder, E. Pinheiro, and W. Weber. DRAM Errors in the Wild: A Large-scale Field Study. In *Proceedings of ACM SIGMETRICS*, 2009.

[45] A. Shaikh, C. Isett, A. Greenberg, M. Roughan, and J. Gottlieb. A Case Study of OSPF Behavior in a Large Enterprise Network. In *ACM SIGCOMM WIM*, 2002.

[46] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar. Making Middleboxes someone else's Problem: Network Processing as a Cloud Service. In *Proceedings of SIGCOMM*, 2012.

[47] C. Talbot. Dropbox Outage Represents First Major Cloud Outage of 2013. `http://goo.gl/rszmb`, January 2013.

[48] D. Turner, K. Levchenko, A. Snoeren, and S. Savage. California Fault Lines: Understanding the Causes and Impact of Network Failures. In *ACM SIGCOMM CCR*, 2010.

[49] M. Wilk and R. Gnanadesikan. Probability Plotting Methods for the Analysis for the Analysis of Data. *Biometrika*, 55(1), 1968.

[50] S. Works. Hurricane Sandy - AC2 Transatlantic Cable Cut. `http://goo.gl/dywVO`, October 2012.

[51] Z. Yin, X. Ma, J. Zheng, Y. Zhou, L. Bairavasundaram, and S. Pasupathy. An Empirical Study on Configuration Errors in Commercial and Open Source Systems. In *Proceedings of ACM SOSP*, 2011.