A Study of End-to-End Web Access Failures

Venkata N. Padmanabhan[†] Sriram Ramabhadran[§] Sharad Agarwal[†] Jitendra Padhye[†]
[†]Microsoft Research [§]UC San Diego

ABSTRACT

We present a study of end-to-end web access failures in the Internet. Part of our characterization of failures is based on directly observable end-to-end information. We also present novel analyses that reveal aspects of end-to-end failures that would be hard to discern otherwise. First, we combine end-to-end failure observations across a large number of clients to classify failures as server-related or client-related. Second, we correlate failures attributed to a client or server with BGP churn for the corresponding IP address prefix(es), to shed light on the end-to-end impact of BGP instability.

Our study is based on failure observations during a month-long experiment involving 134 client hosts (across PlanetLab, commercial dialup and broadband ISPs, and a corporate network) repeatedly accessing 80 websites. We find that the median failure rate of web accesses is about 1.5%, which is non-negligible. About 34-42% of the web access failures are due to DNS problems, primarily due to the inability of the client to connect to its local DNS server. The majority of the remaining failures are due to TCP connection establishment failures. Also, by correlating failure observations across clients and servers, we find that server-side problems are the dominant cause of TCP connection failures.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Misc.

General Terms

Measurement, Reliability

Keywords

Web access, web failure, TCP, DNS, HTTP, BGP

1. INTRODUCTION

We present a study of *end-to-end* failures of web accesses in the Internet. A web access consists of a client downloading one or more objects from a web server. The access could fail in different ways (e.g., at the DNS, TCP, or HTTP levels) and for a variety of reasons (e.g., problems with the access link, local DNS, WAN connectivity, website, etc.). Our objective here is to develop techniques for and to present a characterization of such end-to-end failures. The client vantage point reveals a more complete picture of end-to-end failures than monitoring of any individual component would, albeit only for the (limited) set of clients that we are in a position to monitor. So we believe that our work complements prior work focused on the server-side view (e.g., [8, 9, 21]), which provides an incomplete picture of a much broader set of clients. It also complements work focused on a detailed analysis of individual components of end-to-end communication (e.g., DNS [18, 22, 23], BGP [19, 12, 14, 26]) or traceroute-based fault analysis of the IP path [32, 16].

We start with analysis that is based on information directly available from the individual web accesses observed at each client. In particular, we classify web access failures as being due to DNS (inability to resolve the website name), TCP (inability to do a TCP transfer from the server to client), or HTTP (inability of the server to return the requested content). DNS and TCP failures can further be classified into unresponsive local DNS server, TCP connection establishment failure, etc.

We then turn to gaining a deeper understanding of the nature of end-to-end failures by tapping into information beyond that obtained from observing individual web accesses at each client. We present two novel analyses.

First, we combine failure observations made across clients and web servers to identify the extent of correlation in the failure patterns. Such correlation, or the lack thereof, is used to infer the likelihood of an end-to-end failure being due to a client-side problem (i.e., affecting a significant fraction of a client's communication with various servers), a server-side problem (i.e., affecting a significant fraction of a server's communication with various clients), or otherwise. Such a determination would be hard to make based just on individual web accesses.

Second, we determine the extent to which client-side or server-side problems coincide with network routing instability at the inter-domain level. We identify the latter based on BGP churn in the corresponding IP prefixes. Our goal is to understand the relationship between network routing problems and end-to-end failures.

We present measurements from a month-long experiment conducted in Jan 2005, in which a set of 134 client hosts repeatedly accessed a diverse set of 80 websites. The clients were distributed geographically (although the majority were in the U.S.) and across PlanetLab, the MSN dialup network, multiple residential broadband networks, and the worldwide corporate network of a major corporation. (We are making

our measurement data available online [2].)

We find that the overall failure rate of web accesses is low but non-negligible. The median failure rate across clients is 1.47% and that across servers is 1.63%, representing less than two 9s of availability. So we believe that it is important to understand the nature of these failures. Here are some of the key findings from our analysis of these failures:

- 34-42% of web access failures are due to DNS problems, about 74-83% of which are caused by the client's inability to connect to its local DNS server.
- The remaining failures (57-64%) are almost all due to TCP connection failures. The majority of these are TCP connection establishment failures (i.e., failed SYN handshake).
- Our correlation analysis reveals (somewhat surprisingly) that server-side problems are the dominant cause of TCP connection failures. This is primarily because client connectivity problems manifest themselves as DNS resolution failures, precluding even a TCP connection establishment attempt.
- Although the incidence of failure across websites is highly skewed, 70% of the websites in our study experienced at least one server-side failure episode, which we define as a failure rate of $\geq 5\%$ over a 1-hour period.
- Severe BGP instability for a client or server's IP prefix often implies significant failures for the client or server's end-to-end communication. However, such severe BGP instability is rare and does not account for the vast majority of end-to-end failures.

Besides these and other specific findings, we believe that a key methodological contribution of our work is in correlating failure observations across end hosts to analyze the nature of *end-to-end* failures. We believe that this approach is novel and provides a useful complement to prior traceroute-based approaches (e.g., [32, 16]), especially in settings where firewalls or other filters impede traceroute functionality.

The rest of this paper is organized as follows. We present our analysis framework in Section 2, and our experimental setup and methodology in Section 3. This sets the stage for the presentation of our results and analyses in Section 4. We discuss the implications of our findings in Section 5 and related work in Section 6. We conclude in Section 7.

2. FAILURE ANALYSIS FRAMEWORK

In this section, we present a framework for client-based characterization and analysis of end-to-end web access failures. Part of our characterization is based on information that is directly available from the individual web accesses observed at each client. We also consider inferences that can be drawn by combining end-to-end observations across clients and using indications of network routing instability.

Note that "failure" does not imply a total inability to communicate, but rather just noticeably abnormal behavior (e.g., a failure rate of 15% is abnormally higher than the normal failure rate 1%). We use the terms "failure", "fault" and "problem" interchangeably.

2.1 Failure of Individual Transactions

We begin by discussing the categorization of failures of individual web accesses, or *transactions*. A web transaction consists of a client resolving a web server name to an IP address, establishing a TCP connection to it, and downloading the object of interest using HTTP. We refer to the download of each individual web object as a separate transaction.

A transaction fails when any of these steps fails. These steps proceed in order and the client can tell which, if any, has failed. Thus there are three basic categories of failures directly observable at the client, each of which can be further categorized into sub-classes:

1. **DNS:** The website name cannot be resolved. This can be due to several observable reasons:
   a) Local DNS Server (LDNS) timeout: LDNS is unreachable, because it is down or because of network connectivity problems between it and the client.
   b) Non-LDNS timeout: LDNS is responsive, but the lookup still times out, because of an unreachable authoritative server elsewhere in the DNS hierarchy.
   c) Error response: An error is returned because the name could not be resolved (e.g., NXDOMAIN).
2. **TCP:** Name resolution is successful, but either the TCP session could not be established or unexpectedly terminated. We can observe the following :
   a) No Connection: The client cannot connect to the server, i.e., the TCP SYN handshake fails, either because of a network connectivity problem or because the server is down.
   b) No response: The client establishes a connection and sends its request, but does not receive a response. A server overload or failure of the server application can cause this. While there may be network connectivity issues, the success of the SYN handshake makes it less likely that there was a total connectivity failure.
   c) Partial response: The client receives only part of the server's response before the connection terminates prematurely, either because of a server failure or because of a server/network problem that makes the connection so slow that the client times out and terminates the connection.
3. **HTTP:** The TCP transfer is successful, but the server does not supply the desired content and instead returns an HTTP error (e.g., file not found). Since HTTP failures are rare in our study (under 1-2% of all failures), we do not categorize them further here.

2.2 Correlating Across Clients & Servers

Local observations at an individual client of its communication with a particular server may not always indicate the nature of the problem that is causing the failure. For example, in the case of a "no connection" failure, it is not clear whether the cause is a connectivity problem at the client end, a server failure, or a problem in the interior of the network. Client-based traceroute is impeded by firewalls (as with the corporate clients in our study), does not reveal failures in the server→client direction, and is often incomplete even when end-to-end web communication is successful.

Instead, we disambiguate the likely cause of failure by correlating failure observations across clients and servers. First, we identify *failure episodes*, which are periods with an abnormally high failure rate for a client or a server (compared to the system-wide "normal" behavior, as discussed in Section 4.4). Second, by combining failure observations across clients and servers, we categorize failure episodes as:

1. **Server-side:** If a server is experiencing an abnormally high aggregate failure rate in its communication across many clients, we term the corresponding period as a

server-side failure episode for this particular server. Note that the underlying cause could be a network problem that affects accesses to the server from many clients rather than a problem at the server itself (e.g., BGP instability in the corresponding prefix).

For websites with multiple replicas, a server-side failure episode could affect all replicas (**total replica** failure episode) or affect only a subset of the replicas (**partial replica** failure episode). Note that "total" and "partial" only refer to the spatial extent of the failure episode across the replicas, not to total or partial failure of accesses to the website. So, for instance, an abnormally high failure rate of 20% that affects all replicas of a website would still be termed as a total replica failure episode.

2. **Client-side:** If a client sees an abnormally high aggregate failure rate in its communication across many servers, we term the corresponding period as a *client-side* failure episode for this client.

3. **Client-server-specific:** If a specific client-server pair is experiencing an abnormally high failure rate, but neither the client nor the server is experiencing an abnormally high failure rate in aggregate, then we term the corresponding period as a *client-server-specific* failure episode.

4. **Proxy-related:** If the client's accesses through a particular proxy exhibit an abnormally high failure rate, we label the corresponding period as a *proxy-related* failure episode. Note if all of the accesses of co-located clients go through the same proxy, then it would be hard to tell apart a proxy-specific problem from a separate client-side problem.

5. **Other:** Besides the failure episodes note above, there may be intermittent or transient failures that are not significant enough in intensity to be registered as abnormal for a client, server, proxy, or client-server pair.

Note that while the above categorization may be suggestive of the location of the problem, it does not indicate the root cause with certainty. Also, the categorization is not mutually exclusive. For example, a server-side failure episode could overlap in time with a client-side failure episode. So communication between the corresponding client-server pair would be affected by both.

We defer the discussion of our analysis of BGP instability and its impact on end-to-end failures to Section 4.6.

3. SETUP AND METHODOLOGY

3.1 Overview

We ran our experiment during a one-month period: Jan 1–Feb 1, 2005. During this period, each client host repeatedly accessed a set of URLs, accessing each URL about 4 times per hour. We randomize the sequence of accesses to avoid systematic bias. To limit network load, we download only the top-level "index" file for each web page. The download attempt is terminated (and declared as having failed) if the underlying TCP connection idles (i.e., makes no progress) for 60 seconds; note that the download could take longer provided it does not idle for 60 seconds.

For each download, we record several pieces of information: the DNS lookup time (or failure indication), the download time (or failure indication), and a packet-level trace of the entire transaction. This data is available to the client without requiring additional network communication. In addition, we have the client invoke a DNS lookup using iterative queries to resolve the website's name (starting with the LDNS server, and then working down from the root servers).

Note that the end-host vantage point is crucial to build a full picture of end-to-end failures. Monitoring traffic from a different vantage point — say within the network or at the server — runs the risk of missing certain end-to-end failures (e.g., DNS lookup or TCP SYN failures due to a local fault).

3.2 Clients

We used the 4 sets of clients listed in Table 1:

PlanetLab (PL): We picked 95 PlanetLab nodes across 64 sites. Having multiple nodes at many of the sites enabled us to identify failures that were likely to be client-site-wide. All nodes ran Linux kernel version 2.6.8.

Dialup (DU): We had 5 clients, all located in Seattle, dial into 26 PoPs spread across 9 U.S. cities in the MSN network. The PoPs we picked in each city were operated by different providers, from whom MSN buys service. The clients dialed into the various PoPs in random order and then downloaded the URLs from the designated set also in random order. Thus although we only had 5 dialup clients, we effectively had 26 "virtual" clients, each of which connected to the Internet via a different path and hence provided a different perspective on the wide-area network. All nodes ran Microsoft Windows XP.

CorpNet (CN): We had 5 nodes, labeled SEA1, SEA2, SF, UK, and CHN, across 4 locations on Microsoft's corporate network. All external web requests from each of these 5 nodes were per-force routed via separate HTTP proxy caches. The proxy was located at the local site in all cases except for CHN, where it was located in Japan. In addition, we had another node in Seattle (SEAEXT) that was located outside the corporate firewall/proxy but shared the same WAN connectivity as SEA1 and SEA2. The CN nodes ran various flavors of Microsoft Windows (2000, XP, 2003).

Broadband (BB): We had 7 residential broadband clients (5 DSL and 2 cable modem) spread across 4 ISP networks (Roadrunner, SBC/Yahoo, Speakeasy, and Verizon) in 4 U.S. cities. The access link speed for these hosts was 768/128 (down/up) Kbps or higher.

Our choice of a diverse set of clients (in terms of geographic location and the nature and speed of connectivity)is motivated by the desire to obtain a broader understanding of Internet behavior than can be obtained from the PlanetLab nodes alone, which are predominantly located at academic sites [10]. Although we had a total of $95 + 5 + 6 + 7 = 113$ client machines, the DU clients dialing in to 26 PoPs effectively gave us a total of 134 clients.

While dialup might be on the wane, it remains an important network access technology with a significant presence (e.g., [6] indicates that 30% of U.S. home users were on dialup as of June 2006). Also, for the purposes of our study, the dialup clients provide visibility into *failures* observed on paths through commercial ISPs, and many of these failures are likely to be independent of the low speed of dialup.

Finally, since we were constrained to locate all the dialup clients in Seattle, there is the concern that the extra latency incurred in dialing into remote PoPs might skew the performance numbers. However, given our focus on failure rates rather than absolute performance numbers, this was not a significant concern.

Category	PlanetLab (PL)	Dialup (DU)	CorpNet (CN)	Broadband (BB)
# Clients	95	5 (26 PoPs)	5(+1)	7 (5 DSL, 2 Cable)
Details	US-EDU (50), US-ORG (19), US-COM (4), US-NET (5), Europe (13), Asia (4)	Boston(ILQ), Chicago(ILQ), Houston(ILQ), New York(IQU), Pittsburgh(ILQ), San Diego(ILQ), San Francisco(ILQ), Seattle(ILQ), Wash. DC(IL)	San Francsico (1), Seattle (2+1), UK (1), China (1)	Pittsburgh (1), San Diego (2), Seattle (3), San Francisco (1)

Table 1: Clients used in our experiment. The DU client providers are ICG(I), Level3(L), Qwest(Q), and UUNet(U).

US-EDU (8): berkeley.edu, washington.edu, cmu.edu, umn.edu, caltech.edu, nmt.edu, ufl.edu, mit.edu
US-POPULAR (22): amazon.com, microsoft.com, ebay.com, mapquest.com, cnn.com, cnnsi.com, webmd.com, espn.go.com, sportsline.com, expedia.com, orbitz.com, imdb.com, google.com, yahoo.com, games.yahoo.com, weather.yahoo.com, msn.com, passport.net, aol.com, nytimes.com, lycos.com, cnet.com
US-MISC (15): latimes.com, nfl.com, pbs.org, cisco.com, juniper.net, ibm.com, fastclick.com, advertising.com, slashdot.org, un.org, craigslist.org, state.gov, nih.gov, nasa.gov, mp.com
INTL-EDU (10): iitb.ac.in, iitm.ac.in, technion.ac.il, cs.technion.ac.il, ucl.ac.uk, cs.ucl.ac.uk, cam.ac.uk, inria.fr, hku.hk, nus.edu.sg
INTL-POPULAR (15): amazon.co.uk, amazon.co.jp, bbc.co.uk, muenchen.de, terra.com, alibaba.com, wanadoo.fr, sohu.com, sina.com.hk, cosmos.com.mx, msn.com.tw, msn.co.in, google.co.uk, google.co.jp, sina.com.cn
INTL-MISC (10): lufthansa.com, english.pravda.ru, rediff.com, samachar.com,chinabroadcast.cn, nttdocomo.co.jp, sony.co.jp, brazzil.com, royal.gov.uk, direct.gov.uk

Table 2: The list of 80 web sites that were targets of our download experiment. For the sake of brevity, we have left out the "www" prefix for most of these hostnames. We only downloaded the top-level "index" file at each site.

3.3 Web Sites

We picked a set of 80 websites as the target for our download experiments. As indicated in Table 2, we tried to ensure significant diversity among the web sites in terms of the geographic location, popularity, etc. (Popularity was determined based on the Alexa list [1].) Some of these sites were replicated or served via CDNs.

The number of websites chosen was constrained by the frequency with which each client could perform downloads, without generating excessive network traffic or triggering alarms. In our experiment, each client accessed each website approximately 4 times an hour, which translates to $80 * 4 = 320$ downloads per hour from each client (although the number of TCP connections attempted was higher because of HTTP redirects and also retries by our `wget` client).

Note that for the remainder of this paper, we use the term "server" to refer to the website listed in a URL, and the term "replica" to refer to a specific server IP address.

3.4 Download Procedure

We used off-the-shelf tools to do our measurements. In each measurement iteration, the URLs were sorted in random order. The procedure for each download was:

1. Flush the local DNS cache.
2. Use `wget` to download the URL ("index" file only).
3. Use iterative `dig` to traverse the DNS hierarchy.
4. Use `tcpdump` or `windump` to record a packet-level trace of the entire transaction.

There are also special steps for the DU and CN clients. To minimize the overhead of dialing out in the case of the DU clients, we dial in to a PoP at random and download all the URLs (in random order) at a stretch, before switching to a different PoP. For the CN clients, we configure `wget` to issue requests with the "no-cache" cache-request-directive [15] set, to ensure that the response is received from the origin server. We do so to avoid having the proxy cache mask failures beyond the proxy. However, since the proxy rather than the CN client does name resolution, and there is no way for the client to force the DNS cache at the proxy to be flushed, some DNS failures may be masked from the client.

We did not gather packet-level traces on the BB clients. Since these were users' home computers, there were privacy and storage requirement concerns. Also, while we did gather packet-level traces for the CN machines, these were not interesting since they only revealed the dynamics of TCP connections to the local proxy.

3.5 Post-Processing

From the raw data recorded for each download, we obtain an indication of the success/failure of both the DNS lookup and the download, the DNS lookup time, the download time, and the failure code, if any, reported by `wget`. We store this information in a performance record, together with the client name, URL, server IP address, and time.

We also post-process the `tcpdump/windump` packet traces to determine (a) the cause of connection failure (i.e., no connection, no response, or partial response, as discussed in Section 2.1), and (b) packet loss count (inferred from packet retransmissions).

3.6 BGP Data

To determine in which period a client or server experienced failures that coincided with inter-domain network routing instability, we examine publicly available BGP routing data, from the Routeviews project [4]. We use BGP updates stored in the MRT format from the month of January 2005 from the Routeviews2, EQIX, WIDE, LINX and ISC servers. In total, these 5 servers have 73 peering sessions with a variety of ASes, including several large ISPs such as AT&T, Sprint, and UUNet. The 203 client and replica IP addresses that we consider [1] are covered by 137 BGP prefixes [2], 132 of which are announced from at least 71 peering sessions (or neighbors). The remaining 5 prefixes have very poor connectivity and can be reached from less than 13 neighbors. We processed this MRT update data to obtain the number of BGP route withdrawals and number of BGP route announcements heard for each client or server prefix in each 1-hour episode. We also calculated how many of the 73

[1]We exclude IP addresses that saw too few connections, e.g., server IPs that did not qualify to be a replica, as discussed in Section 4.5.

[2]Of the 203 client and replica addresses, 153 can be reached from only 1 prefix, while the remaining 50 are covered by 2 prefixes. We consider both prefixes in the latter case, to cover the scenario where the more specific prefix has been withdrawn, or worse, filtered by some ASes due to a prefix length filter.

Category	Trans.	Failed Trans.	Conn.	Failed Conn.
PL	16,605,281	458,692 (2.8%)	21,163,180	539,787 (2.6%)
BB	2,307,855	30,023 (1.3%)	2,849,889	19,408 (0.7%)
DU	381,556	2,622 (0.7%)	471,931	2,343 (0.5%)
CN	1,236,544	10,473 (0.8%)	N/A	N/A

Table 3: Overall transaction and connection counts broken down by client type, with failure rates in parentheses. Connection counts are unavailable for CN because these are masked by the proxy.

peering sessions advertised at least 1 announcement for the relevant prefix, and how many participated in withdrawals.

A caveat with this measurement data is that it can exhibit false route updates due to the collection infrastructure. For example, if one of the Routeviews servers is rebooted or session is reset, each prefix will have additional updates that do not reflect a change due to an actual BGP routing event. We follow prior work in "cleaning" our BGP data [31, 5], basically by estimating and disregarding the volume of updates suspected to be due to resets affecting just the Routeviews servers. For each 1 hour period, if more than $60,000$ unique prefixes (i.e., at least half the routing table) received announcements, we assume a reset occurred. We calculate the average number of unique neighbors that each prefix received an announcement from and subtract that from the count of announcements and count of neighbors participating in announcements from all prefixes during that period. We perform the same calculation for withdrawals.

4. EXPERIMENTAL RESULTS

We start in Section 4.1 by presenting the statistics of transaction-level failures, broken down by client category (PL, DU, CN, BB) and by failure type (DNS, TCP, HTTP). In Section 4.2 and Section 4.3, we present a more detailed breakdown of DNS failures and TCP connection failures, respectively, as observed at individual clients. The nature of DNS failures (e.g., whether the local name server is reachable and responsive) is directly observable at clients. However, the nature of TCP connection failures is often not; for instance, when the TCP SYN from a client goes unanswered, it is not evident where the problem is. So in Sections 4.4 and 4.5, we turn to correlating TCP connection failure observations across clients and servers with a view to classifying failures as client-side or server-side. We consider the impact of BGP instability in Section 4.6, and finally turn to investigating proxy-specific failures in Section 4.7.

Table 3 summarizes the overall transaction and connection failure statistics. Note that the number of TCP connections is typically larger than the number of transactions, because of HTTP redirects and retries by our `wget` client. Also, there were times when individual client machines were down and so were not making web accesses.

4.1 Transaction Failure Analysis

In this section, we present overall failure statistics for web transactions over the month-long data set. A *transaction* is an invocation of `wget` to download a URL.

4.1.1 Overall Transaction Failure Rate

We compute the overall failure rate for each client over all its transactions with all servers, and likewise for each server. The median failure rate over the one-month period across clients is 1.47% and across servers is 1.63%. However, it is as high as 10-20% for some clients and servers; e.g., the 95th%-tile of client failure rate is 10%.

Figure 1 plots the mean transaction failure rate for each category of clients (shown as underlined numbers). It is interesting to note that the mean failure rate is lowest (0.69%) for the DU clients and significantly higher (2.76%) for the PL clients, despite the latter being connected to much higher-speed academic and research networks. We confirmed with the MSN operators that they do not employ any caching proxies, transparent or otherwise, that might shield the DU clients from wide-area network failures. The difference in failure rates may be because the DU clients connect via a commercial dialup service (which presumably strives to provide a good quality of service) whereas the PL clients are part of the experimental PlanetLab network (which suffers from problems such as the permanent failures discussed in Section 4.4.2).

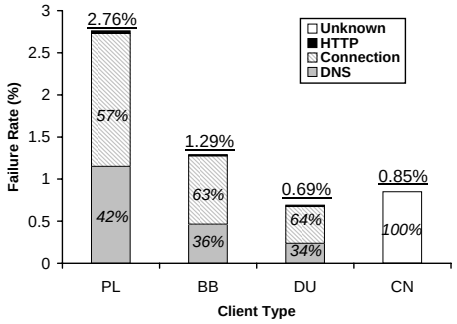

Figure 1: The transaction failure rate, broken down by failure type (in italics) and category of clients. The overall failure rate for each client category appears underlined.

4.1.2 Breakdown of Transaction Failures

Figure 1 also plots the breakdown of transaction failures by type, for each category of clients. (We are unable to provide a breakdown for CN clients, since these connect via proxies that mask the true nature of failures.) The failure types are the ones presented in Section 2.1: DNS, TCP, and HTTP. We find that for all categories of clients, TCP connection-level failures dominate, accounting for 57-64% of all transaction failures. DNS failures account for most of the rest (34-42%). The significant chunk of DNS failures underscores the importance of the end-host view. Observations made from a different vantage point (e.g., a client site's DMZ or the server) would, in general, not reveal these failures.

HTTP-level failures account for under 2% of the transaction failures in all cases. We only accessed the top-level "index" file at each website, which is presumably more available than the average object on the website.

The low incidence of HTTP-level failures in our study contrasts with the finding in [16] that HTTP-level failures constituted about 25% of all failures. There are a couple of reasons for this discrepancy. First, the overall failure rate in [16] was lower because all clients there were on a well-connected university network and because DNS failures were apparently not considered. So HTTP-level failures constituted a larger fraction of the failures in [16]. Second, the majority of the HTTP-level failures in [16] were partial responses, which we classify as "partial response" TCP connection failures, as noted in Section 2.1.

Category	Failure count	LDNS timeout	Non-LDNS timeout	Error
PL	191168	83.3%	9.7%	7.0%
BB	10792	76.0%		24.0%
DU	899	77.7%		22.3%

Table 4: Breakdown of DNS failures

4.1.3 Packet Loss and Transaction Failures

Several previous studies have considered packet loss rate of TCP connections [24, 33]. However, we only find weak correlation between packet loss rate and the failure rate of end-to-end transactions in our data set (the coefficient of correlation is 0.19). We believe this is because: (a) transactions can fail for reasons that have little to do with the end-to-end server-client path (e.g., DNS failures, as shown in Figure 1), (b) a transaction can succeed despite (possibly severe) packet loss, and (c) estimating packet loss rate using TCP traffic is prone to bias, since failed connections that transfer no data (which are, in fact, quite significant, as discussed in Section 4.3) are hard to account for.

Thus we believe that it is important to study the failures of end-to-end transactions rather than only packet loss rate. In the following sections, we analyze the two dominant causes of transaction failures — connection failures and DNS failures — separately. The reason for analyzing these separately is that DNS resolution and TCP/HTTP connections typically involve distinct Internet components and possibly distinct network paths.

4.2 DNS Failure Analysis

We present a breakdown of DNS failures based on the iterative `dig` request that follows each `wget` access. We consider all cases where DNS resolution failed for `wget`. In over 94% of these cases, the interative `dig` also fails; the small discrepancy is due to transient failures.

We see from Table 4 that LDNS timeouts are the dominant cause of DNS failures for PL, either due to the LDNS being down or because of connectivity problems. Non-LDNS timeouts and DNS errors are less common. Unfortunately, due to data collection issues with the DU and the BB clients (which account for a much smaller number of DNS failures than PL), we are not in a position to fully break down the timeout failures for these clients. However, the partial data we have for BB also shows that LDNS timeouts are dominant, accounting for 73.9% of the DNS failures for these clients.

The dominant category of LDNS timeouts represents a client-side failure, whether a last-mile connectivity problem at the client or an unreachable or offline LDNS server. Being so, we would expect LDNS timeouts to affect the resolution of all website names roughly equally. In Figure 2, we plot the cumulative contribution of website domain names to the overall DNS failure count as well as the individual categories of failures (the latter only for PL). The steady slopes of the curves for all DNS failures and the dominant category of LDNS timeouts indicate that indeed these failures do not discriminate across website names.

However, the distribution is more skewed across server domain names in the case of the less common non-LDNS timeout failures and DNS errors (the two curves at the bottom right). For instance, 57% of the DNS errors occur for `www.brazzil.com` and 30% for `www.espn.com`. These are SERVFAIL and NXDOMAIN errors, pointing to buggy or incorrectly configured authoritative servers for these domains. The errors affect a large fraction of the clients, indicating that these are not client-side problems.

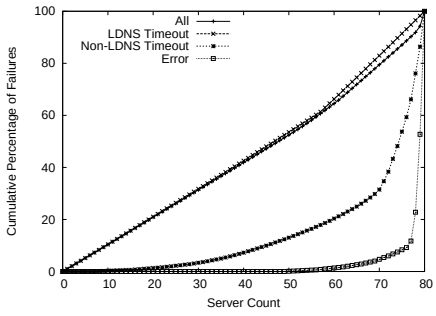

Figure 2: The cumulative contribution of website domain names to the DNS failure count in various categories.

In summary, DNS failures account for a significant fraction (34-42%) of transaction failures. Local DNS timeouts account for a dominant fraction of the DNS failures, pointing to client-side problems (including LDNS failures and last-mile connectivity problems) as the dominant cause.

4.3 TCP Connection Failure Analysis

In this section, we present a categorization of TCP connection failures, which comprise a significant chunk of transaction failures (Figure 1). TCP connection failures are categorized as "no connection" (TCP SYN exchange failed), "no response" (server did not return any bytes of response), or "partial response" (the server returned a partial response, but the connection was terminated prematurely). The breakdown is shown in Figure 3. We see that "no connection" failures dominate in the case of PL (79%) and DU (63%), and are significant in the case of BB (41%). The prevalence of "no connection" failures reinforces the point made in Section 4.1.3 of the unsuitability of TCP packet loss rate as an indicator of transaction failures. It is hard to incorporate information from failed SYN exchanges into an overall packet loss rate metric.

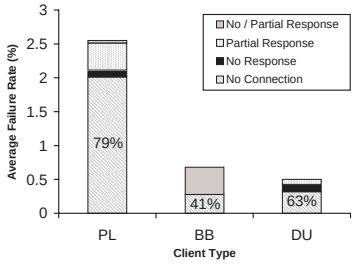

Figure 3: Breakdown of TCP connection failures. The CN clients are excluded since they connect via a proxy, which masks the nature of its wide-area TCP connection failures. The category marked "no/partial response" corresponds to cases where we lacked the `tcpdump` traces needed to disambiguate between the two cases.

TCP connection failures arise either because the server is down (or overloaded) or there is a network connectivity problem between the client and server. To shed more light on the nature of such failures, we now turn to correlating failure observations across clients and servers.

4.4 Correlation Analysis of TCP Failures

As discussed in Section 2.2, we can obtain greater insight into the nature of failures by correlating failure observations across clients and servers. Specifically, we can determine whether failures are associated with a client-side failure episode, server-side failure episode, etc. (Section 2.2). Our goal in this section is to apply such correlation analysis to TCP connection failures.

4.4.1 Classifying Failures

We use a simple *blame attribution* procedure to classify failures. The idea is to associate a set of "entities" with each web access: the client, the server, the client-server pair, the proxy (if any), etc. By aggregating over accesses made across all clients and servers, we compute the failure rate associated with each entity (i.e., each client, server, etc.). We compute these failure rates separately for each episode.

Given a failed web access, we check to see if any of the associated entities has an abnormally high failure rate (defined precisely in Section 4.4.3) associated with it for the corresponding episode. If so, we ascribe the failure to the corresponding entity/entities.

In the remainder of this section, we focus on client-side and server-side failure episodes and present a detailed analysis for these. We briefly discuss proxy-related failures in Section 4.7. But first we consider client-server pairs that experienced near-permanent failure.

4.4.2 Client-Server Pairs with "Permanent" Failures

The distribution of failures rates across client-server pairs is highly skewed: the median is 0.55% but certain pairs experience a transaction failure rate close to 100% over the entire month. 38 out of the 134*80 = 10720 client-server pairs (i.e., about 0.4% of the pairs) experienced a failure rate of over 90% through the month; in 34 of these 38 cases, the failure rate was in fact over 99.6%. The majority of these 38 cases of near-permanent failures involved PL clients and the websites `www.msn.com.tw` (10 cases), `www.sina.com.cn` (9 cases), and `www.sohu.com` (8 cases).

These near-permanent failures appear to be due to a range of causes. In the case of the PL clients at `northwestern.edu` accessing the `www.mp3.com` server, the client starts downloading data but soon encounters TCP checksum errors. However, this problem does not affect other clients when they access this server or the clients at `northwestern.edu` when they access other servers.

A number of clients (e.g., those at `hp.com`, `epfl.ch`, `nyu.edu`, `unito.it`, `postel.org`) encounter near-permanent failures in their accesses to both `www.sina.com.cn` and `www.sohu.com`, which are both websites based in China. Traceroute does not reveal much since it is incomplete and reaches just as far as a traceroute from a client that is able to communicate with this server. It is possible that certain websites are being blocked at particular client sites or that accesses from the affected client sites are being blocked at certain websites.

We defer a more detailed investigation of these near-permanent failures (which would likely involve talking to the concerned IT staff) to future work. To avoid skewing the client-side and server-side failure analysis presented next, we exclude these 38 client-server pairs that could (almost) never communicate. These account for 50.7% of all TCP connection failures but only 13% of transaction failures, the higher contribution to connection failures being due to `wget` retries. With these failures removed, the connection failure rate of PL clients falls to 1.2%.

4.4.3 Identifying Failure Episodes

We need to define the episode duration, i.e. the period over which failure rates for the various entities (the clients and servers, in particular) are computed. There are two conflicting considerations here. We would like the period to be short enough to help identify failures that last say just a few minutes. For example, a 10-minute server outage might stand out on a 1-hour timescale but might be buried in the noise on a 1-day timescale. On the other hand, the period should be long enough for us to have a sufficient number of samples to be able to compute a meaningful failure rate (given the number of clients and servers, and the frequency of accesses in our experiment). To balance both these considerations, we pick 1 hour as the episode duration. We are thus assured a few hundred accesses per client and per server in each episode while also being able to identify relatively short-lived failures. The choice of 1 hour as the episode duration also places minimal requirements on the degree of synchronization needed across the observations made at different clients.

On the flip side, the 1-hour episode duration means reduced resolution compared to intensive probing (for example, [16] probed each path every 15 seconds or more frequently). So, for instance, separate failure events (each lasting say 5 minutes) within the 1-hour period would not be distinguished. This is the price we pay for keeping the access rate of clients and the burden imposed on servers low (well below the level at which our measurements might be noticed and/or elicit complaints).

Next, to decide whether an episode qualifies as a "failure episode" for an entity, we need to determine whether the failure rate for that entity is "abnormally high". Rather than set an arbitrary threshold on the failure rate, we make this determination by comparing with the system-wide normal behavior. Abnormal periods for clients are identified by comparing with all clients and abnormal periods for servers are identified by comparing with all servers. The underlying assumption is that the system as a whole is mostly in the normal state (low failure rate or no failures at all), with abnormal behavior (high failure rate) being the exception. Since we consider failure rates over 1-hour periods, the normal state could well correspond to a non-zero failure rate.

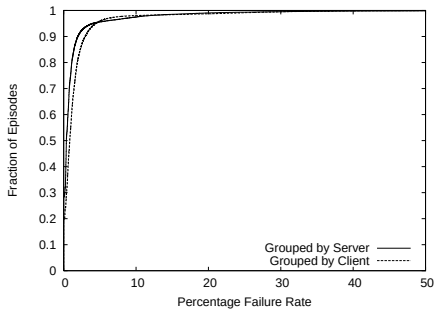

Figure 4: CDF of the overall failure rate over 1-hour episodes across clients and servers.

To identify system-wide normal behavior, we consider the distribution of failure rates, separately for clients and servers, over the 31*24 = 744 episodes in our month-long study (Figure 4). For each 1-hour episode and each client, we compute the failure rate for all of a client's connections across all servers, yielding the "client" CDF shown in the figure. Likewise for servers. We then identify the distinct knee in

Classification	Server-side	Client-side	Both	Other
f=5%	48.0%	9.9%	4.4%	37.7%
f=10%	41.5%	6.7%	0.7%	51.1%

Table 5: Classification of failures for two settings of f.

each CDF that separates the low failure rates (the "normal" range) to the left from the wide range of significantly higher failure rates (the "abnormal" range) to the right. The episode failure rate, f, at the knee is then used to identify the failure episodes, i.e., episodes with an abnormally high failure rate.

In the analysis that follows, we experiment with two settings of the threshold f — 5% and 10% — the latter being more conservative. Although these thresholds might appear to be too low, note that a failure rate of 5% or 10% over a 1-hour episode is actually quite significant for a client or a server, something that users are likely to notice or complain about. Moreover, while the failure rate might have been very high over a short period, the rate would be lower when averaged over a full hour.

4.4.4 Client-side vs. Server-side Failures

We use the procedure outlined in Section 4.4.1 to classify failures as client-side or server-side. We use the threshold, f, to determine whether a failure between a client, C, and a server, S, coincides with a failure episode for C and/or S.

If it is a failure episode only for C, we classify the failure as "client-side". Likewise, if it is a failure episode only for S, we classify the failure as "server-side". If it is a failure episode for both C and S, we classify the failure as being "both" client-side and server-side. If it is not a failure episode for either C or S, we classify the failure as "other", which corresponds to intermittent failures or client-server-specific failures (other than the permanent ones from Section 4.4.2, which we have excluded here).

Table 5 shows the breakdown of failures. Using the two settings of the threshold f — 5% and 10% — we were able to classify 62.3% and 48.9%, respectively, of the failures. It is as expected that more failures would fall in the "other" category when the more conservative threshold of $f = 10\%$ is used to flag failure episodes. We use the $f = 5\%$ setting in later sections, unless stated otherwise.

We see that a significant fraction of failures was classified as "other", because the failures were intermittent and not significant enough to register as significant for either the client or the server on a 1-hour timescale. Of the remaining (i.e., classfiable) failures, the "server-side" category dominates the "client-side" one. In other words, *at the level of TCP connections*, failures are much more likely due to server-side problems (including network problems close to the server) than client-side problems. The fraction of failures that are classified as "both" is small, reflecting the unlikeliness of there being both server-side and client-side problems during the same 1-hour period.

The dominance of server-side failures at the level of TCP connections might seem surprising, given the presumption that large websites are better engineered and managed than client networks. However, there are a couple of reasons why this is so. First, connectivity problems or disconnection at the client end are likely to cause DNS resolution failure and hence preclude even the initiation of a TCP connection. So these would contribute to the DNS failure count (Section 4.2), not the TCP connection failure count.

Second, a server *machine* going offline would cause a large number of clients to experience failures to that server, whereas a client machine being turned off (whether because of problems in our client set or in general because of user actions) would not contribute to access failures because the client would not be making any accesses during the corresponding period. This is just as well since users would only care about failures that happen when they try to access servers and not ones that might happen while their machine is turned off.

4.4.5 Distribution of Server-side Failures

We consider the temporal and spatial distribution of server-side failure episodes. We present data for the $f = 5\%$ threshold; the results for $f = 10\%$ are qualitatively similar.

The total number of 1-hour episodes that were classified as server-side failure episodes was 2732. If we coalesce consecutive failure episodes for a server, the number of coalesced server-side failure episodes is 473. This yields an average coalesced failure episode duration of 5.78 hours. (Recall from Section 4.4.3 that a failure episode means an abnormal failure rate (at least $f = 5\%$ in the present case, not necessarily total failure.) However, the distribution of this duration is highly skewed. The median is 1-hour (the minimum temporal unit used in our analysis) but certain servers suffered from very long failure episodes (e.g., 448 hours at a stretch in the case of `www.sina.com.cn` and 230 hours at a stretch in the case of `www.iitb.ac.in`).

The distribution of server-side failure episodes across websites is also skewed. Table 6 (column 2) shows the large number of 1-hour failure episodes suffered by a small number of servers. For example, `www.sina.com.cn` and `www.iitb.ac.in` suffered server-side failure episodes almost through the month-long experiment. Despite the skewed distribution, a large number of websites suffered server-side failure episodes. During the course of our month-long experiment, 56 out of the 80 websites were affected by at least one server-side failure episode and 39 were affected by multiple failure episodes. So a large fraction of the servers do experience periods of significant failure, even though the overall failure rate is low.

While many of the failure-prone servers were non-U.S.-based (a point we discuss further in Section 4.4.6), some U.S.-based servers also experienced a significant number of server-side failure episodes.

4.4.6 Indirect Validation

It is difficult to directly validate our inferences of server-side and client-side failures, since we have little visibility into the network. Instead, we provide indirect evidence to support our inferences. We do this in two ways.

First, we consider how widespread the impact of server-side failures episodes is, i.e., what fraction of clients is affected in such episodes. We would expect a server-side failure to impact a significant fraction of the clients, and likewise expect a client-side failure to affect transactions to a significant fraction of the servers. The results we present below (see #1) confirm this.

Second, we consider co-located clients (e.g., those on the same university campus) and determine the degree to which their client-side failure episodes are correlated. We would expect a significant degree of correlation, since many failures (though not all) might affect connectivity at the level of the subnet or even the entire campus. The results we present below (see #2) confirm this.

#1: Spread of Server-side Failures

We consider how widespread the impact of server-side failure episodes is. Ideally, we would like to answer this question by looking at how widespread the impact is within each

Server	# server-side failure episodes	Spread
Non-U.S.-based		
sina.com.cn	764	78.4%
iitb.ac.in	759	85.1%
sohu.com	243	72.4%
brazzil.com	97	85.1%
cs.technion.ac.il	95	94.0%
technion.ac.il	90	92.5%
chinabroadcast.cn	89	73.9%
ucl.ac.uk	55	95.5%
U.S.-based		
craigslist.org	166	70.9%
nih.gov	35	60.4%
mit.edu	23	91.8%

Table 6: The list of most failure-prone servers and the "spread" quantifying how widespread the impact of the corresponding server-side failures is.

failure episode. However, this is difficult to do because of sampling limitations.

There are two sampling problems. A server-side problem could cause 100% failure for all client accesses during a short interval, say 10 minutes long. However, there would be no record of failure for clients that happened not to access the server in question during this short period. On the other hand, a server-side problem could last the entire hour but affect say only 20% of the transactions at random. While the underlying problem might be one that does not discriminate between clients accessing the server, there is a chance that some clients get lucky in the sense that none of their accesses to the server fail during the hour. So, in general, we are not in a position to definitively establish which clients *could* have been affected by the server-side problem.

In view of this difficulty, we only look at how widespread the impact of server-side failure episodes is over the entire month-long period.[3] That is, for each server, S, we consider all the failures ascribed to it (i.e., to server-side failure episodes at S). We are interested in how large the set of clients affected by these server-side failure episodes at S over the month-long experiment is. We quantify this "spread" by computing the fraction of all clients needed to account for the failures ascribed to S.

Table 6 lists the spread for the most failure-prone servers. We find that the spread is generally over 70% and often over 80%. (Note that this is much higher than the threshold of $f = 5\%$ used to identify individual failure episodes, with the caveat in footnote 3.) This indicates that the failures that we flag as server-side typically do impact a significant fraction of the clients, as we would have expected. This holds for the U.S.-based servers as well as the non-U.S.-based servers. This serves to indirectly validate the inferences made in Section 4.4.4.

We make one other observation regarding Table 6. Many, though not all, of the most failure-prone servers are located outside the U.S. Given that our client set is dominated by U.S.-based clients, it is hard to distinguish a network connectivity problem between the U.S. and the rest of the world from an actual server-side failure at a non-U.S.-based server that affects a large fraction of the clients. In general, we do not have enough (or any) clients located close to many of the non-U.S.-based servers to be able to tell if such "local" clients were also affected by the apparent server-side failure. However, in some cases we were able to verify that the (small number of) clients located relatively close to the non-U.S.-based server were also affected at the same time that the U.S.-based clients were (e.g., clients in Korea experienced problems accessing `sina.com.cn` and clients in the U.K. experienced problems accessing `ucl.ac.uk`).

[3] Of course, this is not a perfect measure either since multiple distinct server-side problems during different episodes through the month could have affected different subsets of clients. The overall "spread" across clients might be large, but the spread during individual failure episodes could still be small.

	Co-located pairs	Random pairs
# client pairs	35	35
# Pairs with similarity > 75%	2	0
# Pairs with similarity in 50-75%	6	0
# Pairs with similarity in 25-50%	10	1
# Pairs with similarity < 25% & > 0%	10	7
# Pairs with similarity = 0%	7	27

Table 7: The measure of the similarity in the client-side failure episodes experienced by pairs of co-located clients vs. random pairs of clients.

Client pair	# client-side failure episodes in the union	Similarity
planet{1,2}.pittsburgh.intel-research.net	387	98.2%
csplanetlab{1,3}.kaist.ac.kr	5	60.0%
csplanetlab{3,4}.kaist.ac.kr	7	57.1%
csplanetlab{4,1}.kaist.ac.kr	6	50.0%
planetlab{1,2}.comet.columbia.edu	196	3.6%
planetlab{2,3}.comet.columbia.edu	278	52.2%
planetlab{3,1}.comet.columbia.edu	155	5.2%

Table 8: Examples of co-located clients and the similarity in the client-side failure episodes that they experience.

#2: Correlation Between Co-Located Clients

We consider the extent to which client-side failure episodes are correlated across co-located clients. For each pair of co-located clients, we first determined the subset of episodes that were (separately) marked as a client-side episode for each client in the pair. We compute the *similarity* measure for the pair of clients as the ratio of the size of the intersection set (i.e., the client-side failure episodes in common) to the size of the union (i.e., episodes that are marked as a client-side failure episode for either or both clients).

We identify 35 pairs of co-located clients in our data set, most of them being PL clients co-located on the same academic network. But we also had two pairs of co-located BB nodes, one pair each on the Roadrunner cable network in San Diego and the Verizon DSL network in Seattle.

Table 7 shows the similarity measures across the client 35 pairs of co-located clients, and also 35 randomly-paired clients, for comparison. We see that a little over half of the co-located client pairs had a similarity of at least 25%, including about a quarter that had a similarity of at least 50%. In contrast, only one of the randomly paired clients had a similarity greater than 25%. Also, relatively fewer co-located pairs (20%) exhibited zero similarity, about 80% of the random pairs exhibited zero similarity. This indicates that co-located clients exhibit significantly greater sharing of client-side failure episodes than randomly-paired clients. The overwhelming majority of co-located client pairs with low or zero similarity experienced a very small number of client-side failure episodes through the month-long experiment (just 1-2 episodes, in some cases). Any mismatch (i.e., lack of sharing) in these rare failure episodes would result in a low similarity measure.

In general, a low degree of similarity in the client-side failure episodes for co-located clients could also arise because the failure was truly client-specific. One of the examples we consider next illustrates this point.

Table 8 lists a few examples of co-located clients that we

studied. Each set of clients exhibits very different behavior.

The two nodes at Intel see a very large number of client-side failure episodes (387) between them. Moreover, there is a very high degree of similarity (98.2%) across the failure episodes experienced by these two co-located clients. On the other hand, the 3 nodes at KAIST experience only a handful of client-side failure episodes, about 50-60% of which are shared by them.

The case of the Columbia clients is remarkably different. Two of the nodes — #2 and #3 — experience 247 and 192 failure episodes, respectively, with a similarity of 52.2% between them. However, the behavior of the third Columbia node (#1) is very different. It only experiences 12 client-side failure episode, resulting in a low similarity (3.6% and 5.2%) with respect to the other two Columbia nodes.

In summary, we find that a little more than half the pairs of co-located clients shared 25% or more of their client-side failure episodes. Most of the rest were pairs that saw very few client-side failure episodes, making any similarity computation noisy.

4.5 Replicated Websites

We repeat the correlation analysis at the granularity of server replicas to sub-classify the server-side failure episodes as either *total* or *partial* replica failures. As noted in Section 2.2, total failures affect all replicas of a website, while partial failures affect only a subset of the replicas.

We identify the set of replicas for a server S by considering all distinct IP addresses to which connections were attempted by any client while downloading content from S. To make our analysis meaningful, only IP addresses that account for at least 10% of the total number of connections to S are considered to be replicas. As a result, out of the 80 websites used in our experiments, 6 had zero replicas, 42 had exactly one replica, and 32 had multiple replicas. The 6 websites with zero replicas are basically those served by CDNs like Akamai, where the number of distinct IP addresses is very large, so that none of the IP addresses qualify to be counted as a replica per our definition above.

We found that 62% of the failure episodes marked as *server-side* belonged to the 32 servers that had multiple replicas. Of these episodes, an overwhelming majority of 85% were *total* replica failures, which means that *all* replicas of the website were experiencing more than the threshold failure rate during that episode. This is somewhat surprising, but more detailed analysis shows that almost all of the total replica failures are due to websites whose replicas appear to be on the same subnet (same /24 prefix), and hence are prone to correlated failures.

4.6 BGP Analysis

We now consider the relationship between end-to-end connection failures and inter-domain routing instability. We look for BGP instability in the prefix(es) corresponding to each client and server, and consider how these relate to client-side or server-side failure episodes. Clearly, BGP instability that originates close to a client or server has the potential to negatively impact the client or server's wide-area communication more than one that originates further away. We do not consider client-server-specific failures, since each client accessed each server only a small number of times in each 1-hour episode.

If we assume that the most common cause of such failures is temporary, such as a router reboot or session reset, we would expect the outage duration to be at least as long as BGP convergence times. This will likely be in the range of 30 seconds to 15 minutes for a route withdrawal and subsequent route announcement to converge [19]. Thus for a 1-hour period of TCP connection attempts to/from an affected prefix, we expect failure rates in the range of 1% to 25% for temporary outages.

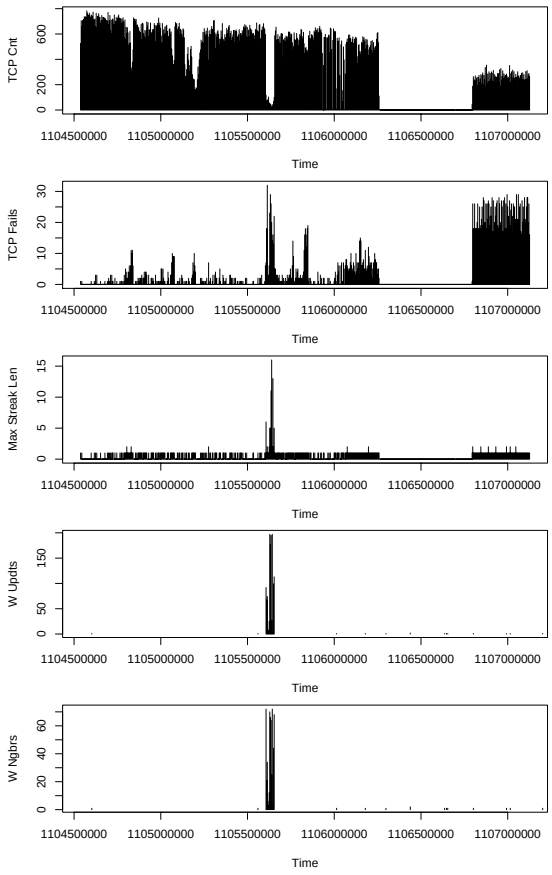

Figure 5: TCP failures and BGP activity for `nodea.howard.edu`, in 1-hour bins. The blank period around 1106500000 corresponds to the client itself being down.

To illustrate the data, we present a particular client `nodea.howard.edu` in Figure 5. The top three graphs show connection data, while the bottom two show BGP data for the client's prefix. The top graph shows the number of TCP connection attempts made in each hour across the entire collection period. We typically attempt about 800 TCP connections per hour, but delays due to low throughput or waiting for timeouts can reduce the number of attempts in a period. The second graph shows how many of these attempts failed to receive a TCP SYN ACK (i.e., "no connection" failure).

One hypothesis we entertain is that during a BGP failure event, remote access attempts will fail consecutively until the client prefix is reachable from all ASes. Thus if we consider the longest consecutive streak of access failures in each 1-hour episode, we should expect it to correspond to BGP events. However, the caveat with this hypothesis is that it assumes BGP convergence takes just as long for all ASes trying to respond to the client. In reality, some ASes will converge on the client prefix faster than others depending on the AS topology [20], and thus some intermediate accesses may succeed while others fail. The third graph in Figure 5 shows the length of the longest consecutive streak of failures in each period.

The bottom two graphs show the BGP data for this client prefix. We plot the number of route withdraws made and the number of Routeviews neighbors (peering sessions) participating in the withdraws. The graphs for announcements are similar, but more noisy.

The graphs clearly show that the period of BGP activity around timestamp 1105632000 is severe in that almost all the 73 Routeviews neighbors withdrew their routes for this client. In fact, multiple announcements and withdrawals were made during this period from each neighbor. We believe this is due to a failure close to the client site, because it is unlikely that multiple failures occurred at the same time in distant parts of the Internet that all affected connectivity to this prefix from all Routeviews neighbors. This period corresponds well to the number of TCP failures in the second graph, and also to the length of consecutive failures in the third graph. There are other periods of high TCP failures but with little BGP activity. However, in these cases the streak length is small, indicating intermittent failures.

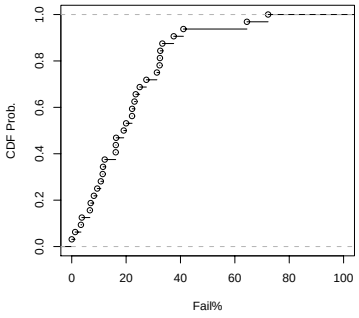

Figure 6: CDF of TCP failures during periods of severe BGP instability ($\geq$75 withdrawals involving $\geq$50 neighbors).

We want to identify periods of BGP instability and correlate them with TCP connection failures. We define a period of BGP instability as a 1-hour period where at least 70 of the 73 Routeviews neighbors withdrew the relevant prefix for a client or replica. If only a few neighbors withdrew the prefix, then it is most likely not a global reachability problem with the prefix, but rather a more local problem with the few neighbors. Across all 719 one-hour periods and 203 clients and replicas considered, there are only 111 instances of BGP instability. The TCP failure rate was over 5% in over 80% of these 1-hour periods.[4] If we use a different definition of BGP instability — at least 50 neighbors withdrawing, but with at least 75 withdraw messages in all — there are fewer periods of such instability (32 1-hour periods compared to 111) but the correlation with TCP failures is stronger. As shown in Figure 6, in almost 80% these periods of BGP instability, the failure rate is over 10%, and in 50% it is over 20%.

However, in the above analysis, we treated all the BGP neighbors equally, although in reality some may impact reachability more widely than others. Consider another client `planetlab1.kscy.internet2.planet-lab.org` in Figure 7. The client experiences a 56% TCP failure rate (129 TCP connection failures out of 227 attempts) around timestamp 1106856000. This corresponds exactly to the spike in BGP withdrawals, and also the neighbors involved in these withdrawals, at the same point in time. However, only 2 neighbors withdrew their routes, as shown in the bottom graph. Nevertheless, these two neighbors had a drastic impact on

[4]While a failure rate of 5% may seem low, note that it is averaged over an entire hour, not just during the instability.

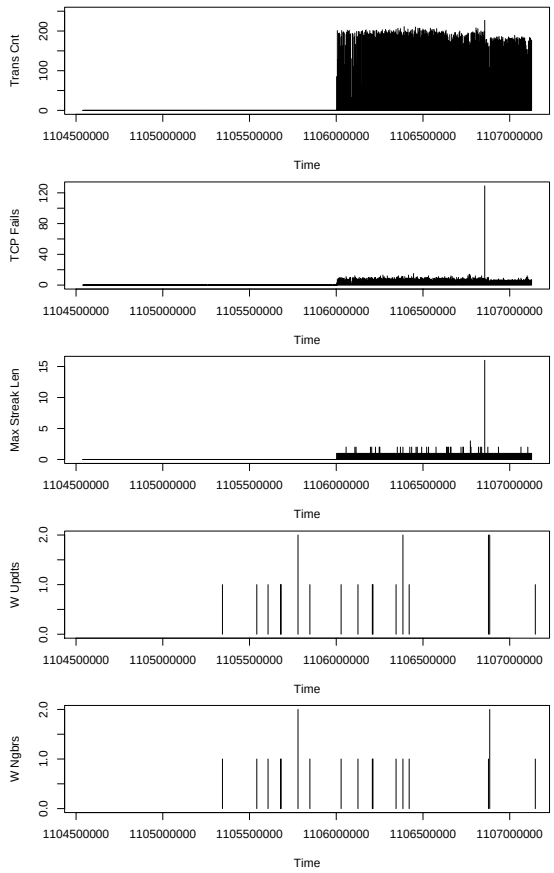

Figure 7: TCP failures and BGP activity for `planetlab1.kscy.internet2.planet-lab.org`, in 1-hour bins. The blank period before 1106000000 corresponds to the client itself being down.

reachability for the client in question. It is likely that most end points (i.e., the websites in our experiment) used these two neighbors for reaching this client.

For our data set, we draw two conclusions here. Firstly, BGP routing events that would impact most web accesses to or from a client or server replica are rare. In less than 0.08% of our data points did at least 70 of the Routeviews neighbors withdraw routes to a client or replica prefix. Secondly, during these rare events, we do find that a significant percentage of TCP connection attempts fail.

4.7 Proxy-Related Failures

As noted in Section 3, our client set includes 5 clients (SEA1, SEA2, SF, UK, and CHN) spread across 4 corporate locations. The web accesses made by these clients are routed through 5 separate caching proxies (Microsoft Internet Security and Acceleration (ISA) servers). In addition, we had a client (SEAEXT) that did not connect via a proxy but shared the same WAN connectivity as SEA1 and SEA2.

Since we only had a single corpnet (CN) client connect through each proxy, it is hard to tell apart client-side and proxy-related failures using the blame attribution procedure described in Section 4.4.1. However, the blame attribution procedure can be used to identify proxy-related failures that are *shared* across all 5 proxies. Such shared failures could arise because of common configuration settings across the proxies (despite the very different locations and wide-area connectivities of the proxies).

	SEA1	SEA2	SF	UK	CHN	EXT	Non-CN
iitb	5.31	5.35	5.33	5.49	5.68	0.23	0.32
royal	6.30	6.21	4.34	7.74	6.94	0.04	1.38

Table 9: Failure rates of accesses (%) to iitb and royal after excluding client-side & server-side failures

We identify two cases of shared, proxy-related failures, pertaining to accesses made to the www.iitb.ac.in and the www.royal.gov.uk websites. For both these websites, we first filtered out failures that were categorized as server-side. Also, for each client, we filtered out all failures of accesses to these two websites that were categorized as client-side. We were then left with failures of accesses to these two websites that were *not* attributable to either a widespread problem at the server end or the client end.

Table 9 shows the rate of these residual failures for each of the CN clients and for the other clients combined. We observe that the residual failure rate for the 5 CN clients that connect via proxies is significantly higher (generally over 5%) than that for non-corpnet clients ("non-CN"). This is so despite the proxies being in different locations with different WAN connectivity. Furthermore, the residual failure rate for the external CN client, SEAEXT, is much lower than that for SEA1 and SEA2, despite all 3 clients sharing the same wide-area connectivity. These suggest that the high failure rate observed by the 5 internal CN clients is because of a shared, proxy-related problem.

On closer inspection, we find that www.iitb.ac.in resolves to 3 IP addresses. Often one or two of these IP addresses is unreachable from all clients, possibly because the corresponding machines are down. However, the failure of all 3 IP addresses is much less common. So *wget* running on a non-proxied client host is often able to connect to the site, by failing over to the alternate addresses if the connection to one address fails. However, the CN proxies do not fail over, presumably to minimize overhead. So if the first IP address that the proxy attempts to connect to is not working at that moment, the client's web access would fail.

At this point we do not have a satisfactory explanation for why the proxied accesses to the other website, www.royal.gov.uk, exhibit a high failure rate.

5. IMPLICATIONS OF OUR FINDINGS

We discuss the implications of our findings for understanding and addressing web access failures.

First, we find that a significant fraction of web access failures are due to DNS problems, and a vast majority of these are due to the inability of the client to access the local DNS server. This implies that improving the reliability of the DNS lookups will go a long way towards improving the overall web browsing experience of the clients. In particular, it is important to address the performance of the "first mile" (i.e. the path between the client and its local DNS server), as well as the reliability of the local DNS server, to reduce the end-to-end web transaction failure rate.

Second, we find that only severe BGP instability for a client's or a server's IP prefix coincides with increased failure rate of end-to-end web transactions. This, coupled with prior findings [7] that general BGP instability does not affect most traffic, suggests that to improve the reliability of end-to-end web transactions, it is more important to address severe episodes of instability that affect multiple peers than general routing instability.

Third, we find that a majority of the connection failures are due to the failure to *establish* the connection, i.e. failure of the SYN handshake. Since the TCP attempts multiple times to establish a connection before declaring failure, such failures indicate a temporary loss of end-to-end connectivity, say due to an overloaded server refusing new TCP connections or due to severe packet loss.

On the other hand, we find that it is rare for a TCP connection to fail after the initial handshake is successful. The TCP connections in our study, like most web transfers, are short. Once a connection is established, it is likely to run to completion unless there is a rapid degradation in network conditions (e.g., severe packet loss or server reset).

Thus, to improve the failure rate of end-to-end transactions, it is important to focus on the episodes of end-to-end connectivity loss. The average packet loss rate experienced by long-lived TCP connections does not reflect the severity of such connectivity loss and so may be of limited utility for quantifying the reliability of end-to-end web transactions. In general, the burstiness of packet loss matters since the loss of multiple SYN or SYN-ACK packets within a short period could prevent TCP connection establishment.

6. RELATED WORK

Many prior studies of Internet performance and failures have considered individual facets of the wide-area network, such as TCP performance, routing (including traceroute analysis and BGP dynamics), and DNS. A few have considered an end-to-end view, as we have done.

During the 1990s, Paxson conducted pioneering studies of Internet routing [24] and end-to-end TCP dynamics [25]. His traceroute-based methodology has inspired much subsequent work on studying network path failures and routing anomalies. In [11], the authors analyze wide-area network failures using traceroutes, to determine their location, duration, and rate. They report a client-server path failure rate of 15 minutes per day (i.e., 1.04%), with near-client, near-server, and interior network failures all contributing significantly. The PlanetSeer system [32] uses traceroutes from multiple vantage points to obtain a fine-grained view of routing anomalies. They find large numbers of short-lived anomalies lasting less than a minute. While these anomalies are interesting, an end-to-end TCP or HTTP transfer (like in our study) could well succeed despite them. In [16], the authors study wide-area path failures and find that most of the failures occur close to clients. We have already discussed (in Section 4.1.2) some of the differences in the specific findings in [16] compared to our study.

Besides traceroute-based analyses, there have also been studies of the instability and failure of Internet routing protocols. In [12, 14], BGP information from routers in diverse locations is correlated to locate the source of Internet routing instabilities. In [19], BGP instability for specific prefixes is introduced artificially and is shown to have a significant impact on end-to-end connectivity and performance. [30] looks at UDP probe performance, including packet loss, delay and out-of-order delivery in both artifical instability and detected instability of BGP. In [27], they detect forwarding anomalies in traffic in the middle of the network using BGP data. [13] examined the correlation between BGP activity and path failures (as determined using custom and ICMP probes) on the 31-node academic RON testbed, and found that the correlation was stronger for failures in the network core than at the edge. Other studies [17] have considered the failure of intra-domain routing protocols such as IS-IS.

There have also been studies of DNS performance. [18] reports that about 36% of DNS lookups emanating from two sites – KAIST and MIT – returned either no answer or an error. This high failure rate is likely because they considered

a filtered stream, i.e., DNS requests that had missed in the LDNS caches at these sites. In [22], the authors report high availability for a broad set of local and authoritative name servers encountered in logs obtained from Akamai. Availability was determined using ping or a query for the A record of one of the DNS root servers. In contrast, [23] reports on the performance of DNS queries issued by a set of PlanetLab nodes for each other's names. They find many instances of DNS slowdown or failure, which they attribute to overload at the local DNS server. This is consistent with our finding that LDNS timeouts dominate.

Several systems have been proposed or are in commercial use for end-host-based network performance monitoring [28, 29, 3]. Our correlation analysis could as well be applied to data from multiple vantage points gathered by such systems.

Our work is inspired by previous work but is distinguished from it in terms of its broad focus on end-to-end failures of web accesses rather than just specific component(s) such as DNS or routing. To this end, we consider the end-host view, which reveals failures that might be hard or impossible to discern from other vantage points. Also, we conduct measurements from a diverse set of clients with dialup, broadband, and corporate connectivity, besides PlanetLab nodes.

7. CONCLUSION

We have presented a characterization of end-to-end web access failures encountered during the course of a month-long experiment in which a diverse set of 134 clients repeatedly accessed 80 different websites. (We are making our measurement data available online [2].)

The overall failure rate is low (a median rate of 1.47% and 1.63%, respectively, across clients and servers) but non-trivial from the viewpoint of making web accesses highly available. 34-42% of the web access failures are due to DNS problems and 57-64% are due to TCP connection failures. The bulk of the DNS failures are local DNS timeouts, pointing to connectivity or other problems at the client end. The bulk of the TCP connection failures are because of unanswered client SYN requests, which by itself does not reveal whether the problem is at the client end or at the server end. By correlating TCP connection failures across clients and servers, we find that these are dominated by server-side problems, although a significant fraction of failures are transient and cannot be categorized as client-side or server-side. Also, we find that while severe BGP instability for a client or server's IP prefix often results in significant end-to-end failures, such instability is rare. We also apply our correlation methodology to find instances of shared proxy-related failures. Finally, we find some client-server pairs (about 0.4% of all pairs) with near-total failure throughout the month.

We believe that our methodology and results on *end-to-end* failures complement previous studies of individual components (DNS, TCP, BGP). The significant incidence of both DNS failures and TCP failures underscores the importance of the holistic view provided by the end-host vantage point. We believe that the idea of correlating failure observations across a large number of end-host vantage points can be fruitfully applied in many settings, including when firewalls or proxies impede traceroute functionality.

Acknowledgements

We thank PlanetLab as well as the following individuals for access to the clients used in this study: Ramana Kompella, Karthik Lakshminarayanan, Yunxin Liu, Ant Rowstron, Rajesh Rao, Gurdev Sethi, Qian Zhang, and Lidong Zhou.

8. REFERENCES

[1] Alexa Web Top 500. http://www.alexa.com/site/ds/top_500.
[2] Dataset presented in this paper (partial). http://research.microsoft.com/projects/NetProfiler/data/.
[3] Keynote Systems. http://www.keynote.com.
[4] Routeviews. http://www.routeviews.org.
[5] www.caida.org/analysis/measurement/recommendations/. routeviewsjan2003.xml.
[6] U.S. Broadband Penetration Report, June 2006. http://www.websiteoptimization.com/bw/0606/.
[7] S. Agarwal, C. Chuah, S. Bhattacharyya, and C. Diot. Impact of BGP Dynamics on Intra-Domain Traffic. In *SIGMETRICS*, 2004.
[8] M. Allman. A Web Server's View of the Transport Layer. *ACM CCR*, Jul 2004.
[9] H. Balakrishnan, V. Padmanabhan, S. Seshan, S. Stemm, and R. Katz. TCP Behavior of a Busy Web Server: Analysis and Improvements. *INFOCOM*, Mar. 1998.
[10] S. Banerjee, M. Pias, and T. Griffin. The Interdomain Connectivity of PlanetLab Nodes. *PAM*, April 2004.
[11] B. Chandra, M. Dahlin, L. Gao, and A. Nayate. End-to-end WAN Sevice Availability. *IEEE/ACM ToN*, April 2003.
[12] D.-F. Chang, R. Govindan, and J. Heidemann. The Temporal and Toplogical Characteristics of BGP Path Changes. In *ICNP*, November 2003.
[13] N. Feamster, D. Andersen, H. Balakrishnan, and M. Kaashoek. Measuring the effects of internet path faults on reactive routing. In *SIGMETRICS*, 2003.
[14] A. Feldmann, O. Maennel, Z. Mao, A. Berger, and B. Maggs. Locating internet routing instabilities. In *SIGCOMM*, 2004.
[15] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, June 1999.
[16] K. Gummadi, H. Madhyastha, S. Gribble, H. Levy, and D. J. Wetherall. Improving the Reliability of Internet Paths with One-hop Source Routing. In *OSDI*, 2004.
[17] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an IP backbone. In *IMW*, 2002.
[18] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS Performance and the Effectiveness of Caching. *IEEE/ACM ToN*, 10(5), October 2002.
[19] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. *SIGCOMM*, 2000.
[20] Z. M. Mao, R. Bush, T. Griffin, and M. Roughan. BGP Beacons. In *IMC*, 2003.
[21] V. N. Padmanabhan, L. Qiu, and H. Wang. Server-based Inference of Internet Link Lossiness. In *INFOCOM*, 2003.
[22] J. Pang, J. Hendricks, A. Akella, B. Maggs, R. Prisco, and S. Seshan. Availability, usage, and deployment characteristics of the domain name system. In *IMC*, 2004.
[23] K. Park, V. Pai, L. Peterson, and Z. Wang. CoDNS: Improving DNS Performance and Reliability via Cooperative Lookups. In *OSDI*, 2004.
[24] V. Paxson. End-to-End Routing Behavior in the Internet. *IEEE/ACM ToN*, October 1997.
[25] V. Paxson. End-to-end internet packet dynamics. *IEEE/ACM ToN*, 7(3):139–152, June 1999.
[26] D. Pei, M. Azuma, D. Massey, and L. Zhang. BGP-RCN: Improving BGP convergence through root cause notification. *Computer Networks Journal Volume 48, Issue 2*, June 2005.
[27] M. Roughan, T. Griffin, Z. M. Mao, A. Greenberg, and B. Freeman. Combining routing and traffic data for detection of IP forwarding anomalies. *SIGCOMM NeTs workshop*, 2004.
[28] S. Seshan, M. Stemm, and R. H. Katz. Spand: Shared passive network performance discovery. In *USITS*, 1997.
[29] C. R. Simpson and G. F. Riley. NETI@home: A Distributed Approach to Collecting End-to-End Network Performance Measurements. *PAM*, April 2004.
[30] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush. A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. In *SIGCOMM*, 2006.
[31] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Observation and Analysis of BGP Behavior under Stress. In *IMW*, 2002.
[32] M. Zhang, C. Zhang, V. Pai, L. Peterson, and R. Wang. PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In *OSDI*, 2004.
[33] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker. On the Constancy of Internet Path Properties. In *IMW*, 2001.