Inferring Relative Popularity of Internet Applications by Actively Querying DNS Caches[*]

Craig E. Wills
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
cew@cs.wpi.edu

Mikhail Mikhailov
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
mikhail@cs.wpi.edu

Hao Shang
Worcester Polytechnic Institute
100 Institute Road
Worcester, MA 01609
hao@cs.wpi.edu

ABSTRACT

In this work, we propose a novel methodology that can be used to assess the relative popularity for any Internet application based on the data servers it uses. The basic idea is to infer popularity of data servers by periodically "poking" at local Domain Name servers (LDNSs) that service Domain Name System requests from a set of users running Internet applications and determining if LDNSs have cached resource records for the data servers. This approach allows us to measure the relative percentage of pokes that result in a cache hit as a coarse measure of the relative popularity of a particular data server among the users of a given LDNS. In addition, the time-to-live (TTL) of cached DNS resource records can be used to measure the gaps in time when a resource record for a data server is not cached. The cache gaps can be used to infer request interarrivals for more popular data servers.

The methodology can be applied to any Internet application that uses distinguished server names and performs DNS lookups on these names as part of application use. The methodology can be used to collect usage information from any LDNS that accepts DNS queries. As example applications of the methodology, we evaluate the relative popularity of selected Web sites and the relative popularity of different Web servers serving content at a given Web site. We also apply the methodology to servers providing multimedia content, data servers for grid computing, and network game servers. We use data gathered from LDNSs of commercial and educational sites as well as Internet Service Providers serving both commercial and home customers.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Protocols

General Terms

Performance, Measurement

Keywords

Domain Name System, Active Content Measurement

1. INTRODUCTION

Internet applications, such as Web browsers, media players and game software, allow users to access a variety of content available from servers on the Internet. Studies have examined the characteristics of content for these applications, but a difficult question these studies face is understanding if, and how frequently, given content is actually used. Published lists of popular Web sites are one means to identify servers with popular content, but the methodology for determining these lists may not be clear. In addition, these lists only include a small fraction of the popular Web sites on the Internet, not necessarily the sites of interest. Alternately, network traces or proxy logs can be used to obtain real user requests, but these data are often limited to requests from users in a researcher's own organization. Obtaining such data from other organizations is difficult.

In this work, we propose a novel methodology that can be used to assess the relative popularity for any Internet application based on the data servers it uses. The basic idea is to infer popularity of data servers by periodically "poking" at local Domain Name servers (LDNSs) that service Domain Name System (DNS) requests from a set of users running Internet applications and determining if LDNSs have cached resource records for the data servers. This approach allows us to measure the relative percentage of pokes that result in a cache hit as a coarse measure of the relative popularity of a particular data server among the users of a given LDNS. In addition, the time-to-live (TTL) of cached DNS resource records can be used to measure the gaps in time when a resource record for a data server is not cached. The cache gaps can be used to infer request interarrivals for more popular data servers.

The methodology is attractive because it can be applied to any Internet application that uses distinguished server names and performs DNS lookups on these names as part of application use. Due to the pervasive use of DNS, the methodology can be used to answer questions about the relative popularity for applications and data servers of interest.

The weakness of the approach is that it can only infer the relative popularity of application server names for users

---

[*] This work is partially supported by the National Science Foundation Grant CCR-9988250.

IMC'03, October 27-29, 2003, Miami Beach, Florida, USA.

of a given LDNS; it does not measure the precise network activity of all users of an application. Obviously more complete information about application usage can be obtained with logs and packet traces, but these types of information sources are traditionally difficult to obtain from outside of one's local organization. In the case of logs, they may only be applicable to a particular type of application such as the Web.

The methodology can be used to collect usage information from any LDNS that accepts DNS queries to target usage patterns for populations of users based upon the LDNSs chosen to study. Although we used 24-hour data collection in our study, the methodology can also be applied to specific periods during a day or week.

In the remainder of the paper, Section 2 provides a brief description on the DNS mechanism as well as related work. In Section 3 we discuss the methodology in detail including how data are gathered and analyzed. In Section 4 we apply the technique to a log of DNS requests and compare the results of the technique to known requests. Section 5 raises and addresses potential issues of the approach. In Section 6 we describe how we identify different categories of LDNSs for applying the technique. In Section 7 we illustrate the methodology by evaluating the relative popularity of selected Web sites and the relative popularity of different Web servers appearing in traversal links at a given Web site. We also apply the methodology to servers providing streaming content, network game servers, and data servers for grid computing. We summarize the work and discuss its future directions in Section 8.

2. BACKGROUND

2.1 The Domain Name System

The Domain Name System (DNS) is a distributed set of servers primarily used by Internet applications to lookup the network address of a given Internet server [13, 14]. An Internet application needing to lookup a server name first sends a DNS query to a local Domain Name server (LDNS), which is often located at the same site. The LDNS maintains a cache of resource records, such as mappings between server names and IP addresses, called `A` records. DNS also uses `CNAME` records to record canonical names where one name is an alias for another. If the LDNS contains a cached record to satisfy the request then it returns the information to the application. If not then the LDNS contacts a root DNS server to obtain the authoritative Domain Name server (ADNS) for the given resource record and directs a recursive query towards the ADNS. The record is returned to the LDNS, which caches the record along with the authoritative TTL (ATTL) it receives from the ADNS before returning it to the application. The LDNS normally maintains the resource record in its cache until the TTL for the record expires.

2.2 Related Work

Much prior research work related to DNS has been on its performance and its contribution to overall network traffic. The first large-scale study of DNS performance examined DNS traffic at a root name server and found that much traffic was due to bugs and misconfiguration [5]. Another study at a DNS root server also found a majority of bogus queries [2]. A recent study examined the impact of erroneous DNS queries and looked at a number of other performance related issues [9]. That study also examined the impact of varying the TTLs on cache hit rates and found that low-TTLs should not greatly increase DNS-related wide-area network traffic.

Other work has examined the impact of DNS performance for specific applications such as the Web. Shaikh et al found that small TTL values (on the order of seconds) do have a negative impact on latency when DNS is used to select among a set of servers [20]. In our own previous study we found that only 20% of DNS requests are not cached locally and that 20-30% of the non-cached lookups take more than one second [22]. Cohen and Kaplan have proposed proactively refreshing stale cache records to reduce the impact of DNS latency [3].

All of these studies focus on the use and performance of the DNS mechanism and not on the potential of using information in the local DNS server caches to infer usage information. As part of our methodology it is important to identify local DNS servers. This type of identification is related to work in [4] to automatically identify LDNS and ADNSs from graphs of DNS traffic.

3. METHODOLOGY

Our basic methodology is to track the presence of a given server name in the cache of a LDNS. The frequency at which a resource record for the server exists in the LDNS cache is a measure of how frequently the server name is used in a DNS lookup. In the following we describe the methodology for gathering data on the presence of a server name in a LDNS cache and how these data are analyzed to infer relative popularity.

3.1 Data Gathering

Figure 1 is used for reference to describe data gathering and analysis. It shows a timeline of DNS requests for a given server name to a particular LDNS. For discussion, we assume the timeline is in units of minutes and that the ATTL for the server name is 5 minutes. In the figure, DNS requests generated by applications are represented by vertical lines just above the timeline. For a request, the LDNS either responds with information from a cached resource record or, if not present, the LDNS generates a recursive DNS query to obtain an authoritative response for the record, which has an ATTL of 5 minutes. The example shows that the first query occurs at time 2m, which causes the record to be brought into the cache with a TTL of 5 minutes. For the next 5 minutes, all DNS queries are satisfied from the cache. At time 7m, the record expires so that the next query at time 7m30s causes the cache to be refilled and so on.

Under the timeline in Figure 1 are shown what we refer to as "pokes." A poke is a DNS query sent to the LDNS where the query is specified to be non-recursive. We use the tool `dig` to make these queries, but other similar tools could be used. The LDNS responds to a poke in the same way it responds to a normal DNS query except that with the non-recursive bit set, the LDNS reports no answers if a cached record is not available. Use of the non-recursive flag is important because it means our pokes do not pollute the cache. The only way in which the LDNS cache is filled is through requests from other applications.

Because a cached DNS record is small in size, it is expected to reside in the cache for the duration of its ATTL (we discuss the implications if it does not in Section 5).

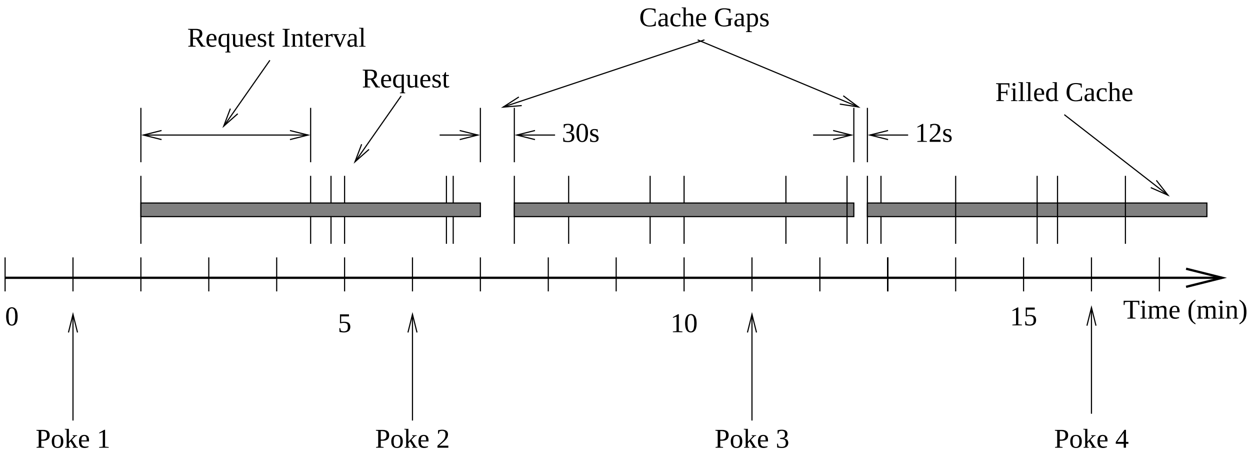

Figure 1: Methodology Description

Therefore, it is not worthwhile to poke at the cache more frequently than the ATTL for a cached record. Thus in the example, we show four pokes spaced 5 minutes apart that arbitrarily begin at time 1m. At each poke we record the time of the poke and if a cached resource record exists in the cache. If it does we also record its current TTL. It is also possible that a response is not received due to a time out in which case we record that an error occurred at the given time.

3.2 Analysis

Two types of analysis can be done on a series of poke results for a given server at a given LDNS. First we can simply determine the cache hit rate over the period of time. The hit rate is the percentage of successful pokes that found a cached record. In the example of Figure 1 this value would be 3 out of 4 or 75%. This percentage provides a coarse measure of popularity that is meaningful for servers not accessed frequently. However, it is less meaningful for servers that are looked up frequently over the duration of the ATTL. This would be the case for popular servers or servers with large ATTLs.

The second type of analysis is to focus on the gap when the cache is empty. These gaps can be computed by using the TTL returned when the cache is poked and the ATTL to determine when the cache is filled with a record and when the record expires. For example, the TTL returned for Poke 2 at time 6m is one minute indicating that the record entered the cache at time 2m and will expire at time 7m. Similar calculations for Poke 3 indicate the record reenters the cache at time 7m30s for a cache gap of 30 seconds. Similarly, the second cache gap in Figure 1 is 12 seconds. In cases where a subsequent poke successfully returns, but does not find a cached record then the gap calculation extends to the next poke. In the case that a poke returns an error or times out then the gap calculation is reset until the record is once again found in the cache.

Intuitively the smaller the cache gap, the more frequently that server is requested, but can we use the gap to infer the request interval? It can be shown that if requests have exponentially distributed interarrival times then measured cache gaps also have the same distribution. This result leads to the question of whether DNS requests exhibit an exponential distribution and in general how well these measured cache gaps approximate the request intervals. In the following section we compare our technique with a known DNS request distribution.

4. COMPARISON OF TECHNIQUE WITH KNOWN DNS REQUESTS

As a means to test whether DNS requests exhibit an exponential distribution and, more importantly, to understand how well the distribution of measured cache gaps can be used to infer the request interval, we obtained a log of DNS requests made to the WPI DNS server (ns.wpi.edu) on our campus. The log was for approximately 28 continuous hours of mid-week activity during April 2003. This server is the primary LDNS for campus as well as the primary ADNS for the wpi.edu domain. We first filtered the log to only consider local DNS requests from WPI clients for non-WPI servers.

We applied our technique of sampling this known request stream with the frequency of the ATTL for numerous servers using the log data. Table 1 and Figure 2 show information about the measured cache gap and the known DNS request interval for three selected servers. Comparative results from these servers are representative of results we found for other servers. We first compare the results for www.google.com. We see in Table 1 that this server uses an ATTL of 5 minutes and that with this sample rate, a cache hit rate of over 92% was found. The table also shows the median, mean and standard deviation of both the measured cache gaps and known request intervals. The results show a larger mean and standard deviation for the cache gap, but comparable medians. These results are reflected in the “gap” and “interval” Cumulative Distribution Functions (CDFs) in Figure 2 where we observe a clear correspondence between the distributions of measured cache gaps and request intervals.

The next comparative results we examine are those for www.yahoo.com. This server uses an ATTL of 30m and at this sampling interval, the server name was found in the cache each time. This server is distinguished in our comparative results because the gap and interval distributions are quite different. Closer inspection of the known requests for this server in the log show that at some points a few clients make DNS requests at regular intervals just a few

Table 1: Cache Hit % and Comparison of Measured Cache Gap with Known DNS Request Interval for WPI DNS Cache

			Cache Gap (sec)			Request Interval (sec)		
Server	ATTL	Hit %	Med	Mean	StDev	Med	Mean	StDev
www.google.com	5m	92.3	11.0	25.5	56.0	8.0	14.3	25.7
www.yahoo.com	30m	100	3.0	3.0	4.1	20.0	32.0	42.3
www.cnn.com	5m	87.8	38.0	42.5	35.6	30.0	52.2	64.7

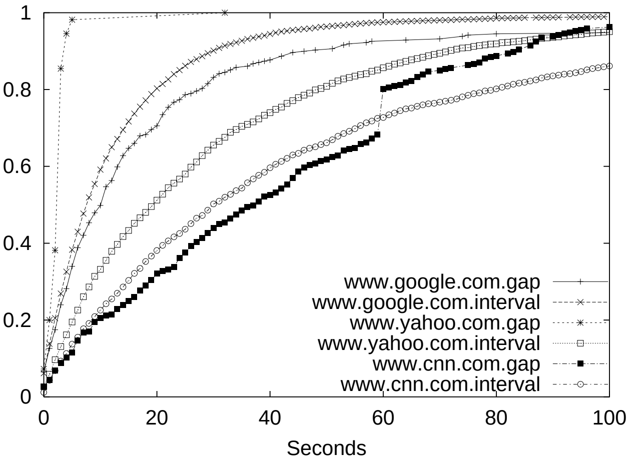

Figure 2: Cumulative Distribution Functions for Measured Cache Gap with Known DNS Request Interval for WPI DNS Cache

seconds beyond the sampling interval of 30 minutes. Over time, particularly during periods of relatively low activity, our pokes "synchronize" with these periodic DNS requests and result in small measured cache gaps even though the request intervals are larger. Applications using these periodic requests are problematic for the technique we propose and represent servers for which it is not appropriate. Fortunately these servers can be identified by looking at their cache gap CDF. In Figure 2 the CDF for cache gaps of www.yahoo.com shows a sharp "corner" at just a few seconds indicating strong periodicity in the underlying request interval. We found such corners or discontinuities in the CDF for a few of the other servers we tested, such as www.weather.com and www.dell.com, where clients generated periodic requests.

The final set of results in Table 1 and Figure 2 are shown for the server www.cnn.com. These comparative results again show a good correspondence between the cache gap and request interval, despite the relatively small amount of discontinuity in the CDF at 60 seconds. Closer examination of the logs shows a short off-peak period where two clients issued DNS requests at 10 minute intervals with six minutes of offset resulting in the measured cache gaps.

In summarizing the comparative results between the measured cache gaps and known request intervals, we make the following observations.

- The request interval distribution is generally not a true exponential distribution.
- While not exponentially distributed, the cache gap is a reasonable estimate of the request interval for most servers, particularly for smaller values of the request interval and cache gap.
- Corners or discontinuities in the CDF of cache gaps reflect periodicity in the DNS request stream and indicate this methodology does not provide good estimates of the true request interval. Fortunately, servers with a strong periodicity in requests, such as what we saw with www.yahoo.com, do not occur often in results we measure.

5. POTENTIAL ISSUES

In addition to the issue of periodic DNS requests, there are other potential issues with the technique. In the following points we identify and address these issues.

- One obvious potential issue with the approach is if the LDNS does not retain the cached record for the lifetime assigned by the ADNS. In our experience with our local WPI DNS server and other LDNSs, we have observed that entries generally do stay for the entire ATTL. In cases where the records do not, the problem can be handled in two ways. First, in the analysis of the pokes, we can observe when this case occurs because we compute a negative cache gap. For example in Figure 1 if the record obtained at time 2m is removed from the cache at time 6m15s (just after Poke 2) then the request at time 6m30s will cause the record to be retrieved again and our calculation for the cache gap at Poke 3 will result in a negative cache gap because the periods that the cache is full appear to overlap. We discard these cases in our analysis and show in Section 6 that they are usually not an issue.

  The second way to detect premature flushing of records from the cache is to poke at the LDNS more frequently than the ATTL. These additional pokes provide no additional information other than to monitor when a record is flushed before its time. We do not use these additional pokes in the results we show, but could do so for LDNSs that are known to flush cached records before the TTL expires.

- A related problem to premature flushing of cached records is when a LDNS does not contain a DNS record for a request and rather than receiving an authoritative response to its recursive query, it receives a cached non-authoritative response from an intermediate DNS server. In previous work [22], we found this occurs in 5-10% of DNS lookups. The implications are similar to the previous issue and we handle it in a similar manner. It might be possible to distinguish between non-authoritative responses from the LDNS and intermediate servers via examination of the response time. We successfully used this technique in [22], but in that work we always communicated with the LDNS via a local area network where in this work, the network latencies to the LDNSs are larger and less consistent.

- Time of day effect, where request intervals and measured cache gaps vary over the course of a day, is another consideration. We found differences over the course of a 24-hour day for all LDNSs in this study. In general, the cache gaps were shorter during "peak" times of each LDNS, but the tone of the results was generally the same. All results reported in this work are for 24-hour days.
- Another issue is how to compare the frequency of access for servers with different ATTLs. The cache gap analysis is independent of the ATTL, but longer ATTLs allow fewer opportunities for sampling and hence fewer observations of the cache gap. It is necessary to find the least common time interval to compare cache hit rates for two or more servers. For example, the hit rates of one server with a 5m ATTL and another server with a 1h ATTL must be compared on an hourly basis.
- The technique can be used to collect usage information from any LDNS that accepts DNS queries from a client. However, finding or gaining access to LDNSs for an organization of interest may be an issue. In practice, while we were not able to gain access to all LDNSs we tried, we were able to gain access to a variety of LDNSs. More details on our approach for identifying LDNSs is given in Section 6.
- In the absence of additional information, it may not be possible to know the population size or characteristics of users served by a LDNS. We can infer characteristics of the population when a LDNS can be traced to a particular location, such as a university campus. We can also infer relative population size by comparing relative frequency of commonly used servers across different LDNSs. We use this technique in Section 6.
- In addition to knowing specific LDNSs to query, the technique requires the server names of interest to be known. These names may be obtained from documentation, examining code or observing DNS queries for the Internet application of interest.
- A potential issue of tracking usage of many servers at a frequent rate at a given LDNS is the potential perception of a denial-of-service (DOS) attack. While possible, the most frequent pokes we used for any server in our study was 5 minutes with pokes for many servers at a much less frequent rate. For about 50 servers tested as part of this work, we generated on average of approximately three DNS requests per minute. In comparison, the WPI DNS server handles over 5000 requests per minute so DOS should not be an issue. Also, in the course of doing this work, no alerts were reported for the DNS traffic we generated.
- The final potential issue of this approach is one of privacy. Just because a researcher has access to cache information of a LDNS does not necessarily mean that the administrator of the LDNS is willing to share that information with others. The privacy concerns of this technique are appropriate to analyze, particularly when the contents of a LDNS cache can be corroborated with other user access information for a relatively small set of users. For this work, we focus on the goal of identifying the relative popularity of Internet servers and hence only identify the type of, but not the specific, LDNSs used in our study.

6. IDENTIFICATION OF LOCAL DOMAIN NAME SYSTEM SERVERS

In this section we describe the approach used to identify the set of LDNSs for application of the methodology. Our goal in identifying LDNSs was to find a number of such servers serving different sets of users. We identified four categories of LDNSs for study, with all of the LDNSs being in the United States. International LDNSs could also be found, but that was not a focus of this work. The following identifies each of the four categories of LDNSs as well as describes the approach we used in obtaining the specific servers within each category.

1. *Commercial sites.* LDNSs in this category are all in the .com DNS domain. These servers were found by using the `dig` tool to obtain the ADNS for a number of companies (both big and small) and then using directed DNS queries to determine if these authoritative name servers also played the role of LDNS for the company. We identified a DNS server as a LDNS if it returned cached resource record information for our queries. In some cases, the DNS server did not have cached information—either because it is not a LDNS or it refused our DNS query. Five LDNSs from both well-known and lesser-known companies were selected and identified as *com1-5* in our results.

2. *Educational sites.* LDNSs for educational sites were identified in a similar manner as for commercial sites using the ADNSs of university sites as a starting point. The selected LDNSs for this category, all from the .edu DNS domain, are located at well-known universities. Five servers, identified as *edu1-5*, are used in our study.

3. *Internet Service Providers (ISPs) serving commercial companies.* In looking for LDNSs of commercial sites, we found a number of ADNSs for these sites being ISPs from the .net DNS domain. In many cases these ADNSs also provide the role of a LDNS. While these ISPs may serve non-commercial customers they were all identified in this category because they serve at least one commercial site. Five of these servers, identified as *ispcom1-5*, are used in our study.

4. *ISPs serving home customers.* The LDNSs in this category were found with a different approach than the other categories. In this case we used published addresses for DNS servers of ISPs known to serve home customers. We found these addresses from help information at the Web sites of the ISPs themselves as well as from technical help information from sites such as [16]. Five of these servers, all from the .net DNS

domain and identified as *isphome1-5*, are used in our study.

To gauge the relative size of the user population for the 20 selected LDNSs, we examined the measured cache hit rate and cache gap for the generally popular server name www.google.com at each of these LDNSs over a one-week period in April 2003. The results for all 20 servers are shown in Table 2. For illustration, Figure 3 provides more detail for the cache gap of the isphome servers with the respective CDFs.

Table 2: Cache Hit % and Measured Cache Gap (sec) of www.google.com for Selected Local DNS Servers

LDNS	Hit %	Med	Mean	StDev
com1	96.7	4.0	11.4	22.1
com2	79.2	26.0	88.1	157.6
com3	80.5	25.0	72.9	116.3
com4	43.7	62.0	378.3	938.6
com5	96.0	7.0	14.2	27.5
edu1	90.4	12.0	32.6	68.6
edu2	90.0	10.0	33.5	76.2
edu3	96.7	5.0	10.9	19.1
edu4	96.8	4.0	19.2	45.1
edu5	97.1	3.0	9.1	22.8
ispcom1	99.9	0.0	1.1	13.4
ispcom2	93.5	8.0	21.1	36.3
ispcom3	98.8	2.0	4.5	7.7
ispcom4	93.0	9.0	23.1	37.8
ispcom5	97.9	4.0	6.9	12.2
isphome1	99.3	2.0	3.2	4.5
isphome2	96.0	6.0	13.0	32.7
isphome3	99.0	1.0	3.4	11.3
isphome4	99.5	49.0	73.2	77.6
isphome5	98.7	2.0	4.8	12.6

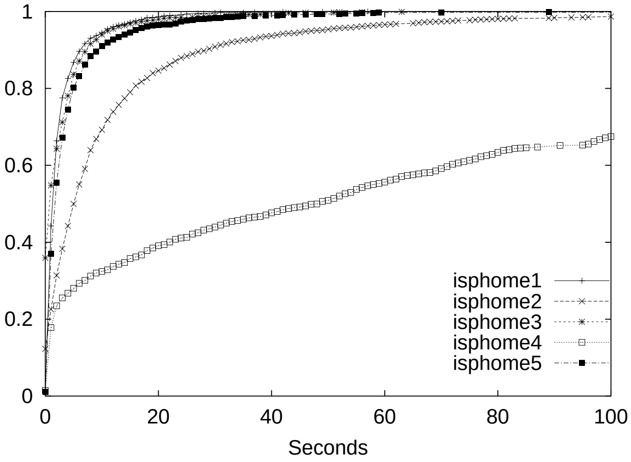

Figure 3: Cumulative Distribution Function for Measured Cache Gap of www.google.com for Home ISP LDNSs

The results show the most variability in relative frequency within the five commercial sites with two of the servers handling more requests than the other three. The LDNSs for the other categories all show frequent accesses with relatively high cache hit rates (using a sampling period of 5 minutes corresponding to the ATTL) and small cache gaps. The busiest LDNS is ispcom1 based both on these results and other results we obtained throughout this study.

We also used these data to determine the prevalence of negative cache gaps in our analysis, which, as described in Section 5, occurs when a record is flushed from a LDNS cache before it expires or the record does not enter the cache with the ATTL for the record. For 17 out of the 20 LDNSs, the calculation of negative cache gaps occurred in less than 1% of the calculations. For the com2 LDNS it occurred in 3% of the cases while it occurred in 7% of cases for edu4. For the isphome4 LDNS it occurred in 40% of the cases indicating a clear problem with prematurely expiring records and the need for more frequent sampling to correct for it. Because we did not introduce more frequent sampling, we do not use results from any of these three LDNSs in the results shown.

7. APPLICATION DOMAINS

The methodology can be applied to any Internet application that uses distinguished server names and performs DNS lookups on these names as part of application use. The methodology is helpful in identifying whether an application is used and if so then to provide data on the relative frequency. This technique can be used to characterize the usage patterns for a targeted community of users and point to what Internet applications and content are appropriate to use for characterization studies.

The technique works best when the DNS lookup of a server name by an application infers meaning about the application use. For example, the lookup of www.mysite.com likely means that the Web site is being visited by a browser or other Web client. Users of network game applications first lookup the name of a game server when joining a game. These lookups can be used to infer the use and relative popularity of Web servers as well as applications.

In this section we apply the technique to study usage patterns for five sample applications. These applications are:

1. *Popularity of Web Servers.* An obvious application is to determine the relative popularity of a set Web servers. Published lists of popular Web sites are one means to identify popular content [18, 17, 15], but the methodology for determining these lists may not be clear and these lists only include a small fraction of Web sites on the Internet. For this sample study we chose to track the relative popularity of a set of Web search servers.

2. *Web site traversal.* An interesting problem for a given Web site is to determine the links that are followed at a site assuming that server logs are not available. This problem arose in previous work by the authors [12]. For sites that use distinct server names for portions of the Web site content, our technique can be used to track relative request frequencies for these servers. In this study we use the CNN Web site (www.cnn.com), which employs many distinct server names for specific types of content at the site.

3. *Streaming content popularity.* A related problem to

Table 3: Cache Hit % and Measured Cache Gap (sec) of Web Search Servers for Selected Local DNS Servers

Server	ATTL	Hit %	1H Hit %	Med	Mean	StDev
LDNS: ispcom1						
www.google.com	5m	99.9	100.0	0.0	1.1	13.4
search.msn.com	1h	100.0	100.0	1.0	3.9	8.3
www.altavista.com	5m	96.0	100.0	3.0	14.4	33.9
search.aol.com	5m	84.2	100.0	21.0	55.9	131.2
www.alltheweb.com	12h	100.0	–	22.5	268.2	763.0
altavista.com	5m	82.7	100.0	23.0	62.1	171.3
www.teoma.com	5m	16.3	85.0	847.0	1469.1	1955.6
teoma.com	5m	2.9	28.1	4905.0	8641.2	10821.5
LDNS: isphome1						
www.google.com	5m	99.3	100.0	2.0	3.2	4.5
search.msn.com	1h	100.0	100.0	5.0	15.5	82.7
www.altavista.com	5m	89.1	100.0	17.0	38.2	63.3
search.aol.com	5m	64.8	100.0	76.5	163.5	272.7
altavista.com	5m	62.5	99.4	89.0	180.7	317.9
www.alltheweb.com	12h	100.0	–	188.0	1423.5	3811.4
www.teoma.com	5m	10.8	68.3	1289.0	2429.4	3397.9
teoma.com	5m	1.7	19.8	13125.5	15848.3	14337.8

determining what Web servers are used is to determine what streaming content available on the Internet is used. In current work to study the characteristics of audio and video streaming content available on the Internet [11], the authors identify which servers store streaming content, but need a means to determine the extent to which it is being used. We apply our technique to a set of servers containing streaming content.

4. *Network games*. Network games are a popular Internet application. We track the use of well-known game servers for specific games to infer the usage patterns of these games.

5. *Grid computing*. The contribution of excess computing capacity to a computational grid is another area of interest. We track the lookups of data servers for two well-known grid computations to infer the usage patterns of contributions to these computing grids.

We gathered data for a set of servers in each of these domains for approximately one-week periods in April and May 2003. While we gathered information from all 20 LDNSs described in Section 6, we show results in this section from only a few LDNSs—primarily from ispcom1 and isphome1. We justify this presentation approach because of space considerations and because the focus of this work is application of the technique, not the results obtained from it.

7.1 Popularity of Web Servers—Search Engines

We choose to test our technique on the popularity of Web search engines because searching is important to many Web users. To identify the list of search servers to track, we largely drew from a list of major search engines [21]. The list of six servers we tested along with their ATTL and any notes about the server name are given below.

1. www.google.com, 5m. A Web request to google.com causes a HTTP redirect to www.google.com.

2. www.alltheweb.com, 12h. A Web request to alltheweb.com causes a HTTP redirect to www.alltheweb.com.

3. www.altavista.com, 5m. Lookups to altavista.com must be tracked separately because separate DNS records are maintained for each.

4. search.msn.com, 1h. This is the default search server in IE.

5. search.aol.com, 5m. This is the search server for AOL.

6. www.teoma.com, 5m. This is a `CNAME` record to teoma.ask.com. Lookups to teoma.com must be tracked separately because separate DNS records are maintained for each.

The notes for these servers are important to understand as the technique works best if we track the most "meaningful" name in terms of DNS accesses. For example, in the case of google, we track www.google.com because any HTTP requests to google.com are redirected at the HTTP level to www.google.com, which causes a DNS lookup of www.google.com. We also found two search engines—altavista and teoma—where neither HTTP redirection nor DNS `CNAME` records are used to redirect to a single server name. These are the only two Web servers for which we found this situation to occur in our study and to be complete we tracked lookups of both names.

In addition to knowing what server names to track, the technique also requires an understanding of how these names are used by an application—in this case a Web browser. Using packet traces while running the Internet Explorer (IE) browser under a Windows operating system and Mozilla under the Linux operating system, we observed that IE does not issue a new DNS request if the elapsed time is less than the TTL of the resource record. We did observe Mozilla issuing requests before the TTL expires, but not for every new page access. In either case, the DNS requests that would generate recursive requests beyond the local LDNS are the same.

Table 3 shows results for the set of Web search servers from two of the busier LDNSs—ispcom1 and isphome1. For easier comparison the servers in the table are roughly ordered based on frequency of access using cache hit percentage and measured cache gap. The cache hit percentage for each server is shown both at the sampling rate of the ATTL and at a sampling rate of one hour. Note that with a 12h ATTL, the www.alltheweb.com server is not included in the one hour hit rate determination. Figures 4 and 5 show the CDFs of the cache gap for the search servers (dropping the less-used altavista.com and teoma.com servers).

The cache hit rate results show some delineation among servers with a 5m ATTL, but that the top six servers are all accessed on an hourly basis. Overall, for users of these LDNSs, www.google.com is clearly the most used, followed by search.msn.com and then www.altavista.com. The search.aol.com and www.alltheweb.com servers show similar results, although servers, such as www.alltheweb.com, with larger ATTLs, allow fewer opportunities for sampling and require longer testing periods to collect sufficient samples. It is also more difficult to compare the cache hit rates of these servers with other servers. For users of these LDNSs, the www.teoma.com server is clearly the least frequently used search server among the set in our study.

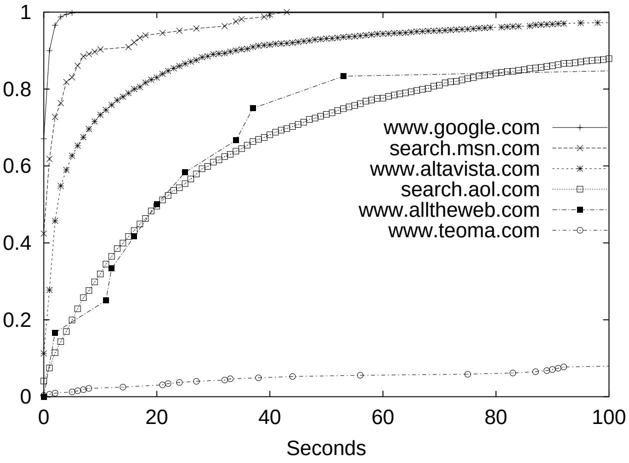

Figure 4: Cumulative Distribution Function for Measured Cache Gap of Web Search Servers on ispcom1

7.2 Traversal of a Web Site—CNN

Another application of the technique is to determine the relative frequency of use for Web sites employing a number of servers to serve portions of the site content. One such site is www.cnn.com where links and forms on the site home page cause requests to a number of servers all under the cnn.com domain. We studied the following eight servers that we found on the CNN home page in April 2003.

1. cnn.com, 5m. A DNS request to www.cnn.com is a `CNAME` record to cnn.com so that cnn.com must be resolved for requests to either server name.
2. search.cnn.com, 1h. This server handles form submissions for search queries.
3. qs.money.cnn.com, 1h. This server handles form submissions for obtaining stock quotes.
4. weather.cnn.com, 1h. This server handles form submissions for obtaining weather information.
5. polls.cnn.com, 1h. This server handles form submissions for taking an online poll.
6. sportsillustrated.cnn.com, 1h. This server handles sports links and is a `CNAME` record to cnnsi.com.
7. money.cnn.com, 1h. This server handles business links and is a `CNAME` record to cnnfn.com.
8. edition.cnn.com, 1h. This server handles the international edition of cnn.com.

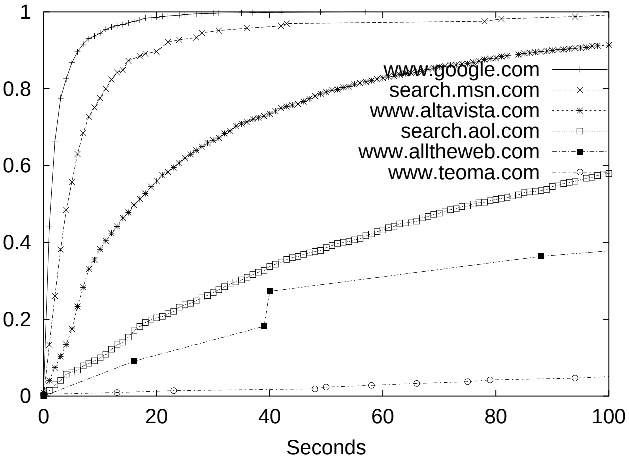

Figure 5: Cumulative Distribution Function for Measured Cache Gap of Web Search Servers on isphome1

Results for these servers with the ispcom1 and isphome1 LDNSs are shown in Table 4 with CDFs for these servers given in Figures 6 and 7. The results show that content from the cnn.com server is looked up relatively frequently by these user populations, but is not as popular as www.google.com. The sports and money servers are the most frequently used other servers with the international edition server showing comparable frequencies for the ispcom1 LDNS, but not isphome1. The four servers used for forms have a lower frequency. Note that as the frequency of use for a server becomes smaller the cache hit rate becomes more meaningful for comparison while the statistics about the measured cache gap become less useful.

7.3 Streaming Content Popularity

While the technique can be used to infer the relative popularity of content available from Web servers, it can also be used to infer popularity of other types of content. In a recent study [11], crawlers were used to find servers serving streaming audio/video content for purposes of characterizing it. An issue with this type of approach is understanding if and at what frequency this content is used.

We applied our technique to a sampling of the servers with the most instances of streaming content found in crawler-obtained data from [11]. We identified the following seven

Table 4: Cache Hit % and Measured Cache Gap (sec) of CNN Site Servers for Selected Local DNS Servers

Server	ATTL	Hit %	1H Hit %	Med	Mean	StDev
LDNS: ispcom1						
cnn.com	5m	98.6	100.0	1.0	4.4	19.1
sportsillustrated.cnn.com	1h	99.4	99.4	5.0	30.7	68.2
edition.cnn.com	1h	97.6	97.6	16.0	102.6	272.3
money.cnn.com	1h	98.8	98.8	18.0	56.8	287.3
weather.cnn.com	1h	89.8	89.8	127.0	400.7	750.3
polls.cnn.com	1h	87.4	87.4	212.0	528.7	877.2
qs.money.cnn.com	1h	69.0	69.0	297.5	1634.5	3800.5
search.cnn.com	1h	71.4	71.4	579.0	1508.1	2636.2
LDNS: isphome1						
cnn.com	5m	94.8	100.0	8.0	17.3	31.3
sportsillustrated.cnn.com	1h	95.8	95.8	61.0	178.9	333.6
money.cnn.com	1h	96.4	96.4	61.5	166.7	306.7
edition.cnn.com	1h	85.1	85.1	374.5	640.1	793.4
weather.cnn.com	1h	79.8	79.8	348.5	895.0	1589.7
polls.cnn.com	1h	83.3	83.3	362.0	735.3	1369.8
qs.money.cnn.com	1h	57.7	57.7	671.0	2619.2	4664.8
search.cnn.com	1h	63.7	63.7	935.0	2095.0	2924.8

Table 5: Cache Hit % and Measured Cache Gap (sec) of Streaming Servers for Selected Local DNS Servers

Server	ATTL	Hit %	4H Hit %	Med	Mean	StDev
LDNS: edu5						
boss.streamos.com	10m	14.7	87.2	1209.0	3449.6	5253.6
mfile.akamai.com	2h	55.2	79.2	2588.0	6010.6	9511.7
www.iuma.com	5m	1.5	48.9	11249.0	16257.7	20114.5
www.connectlive.com	20m	1.7	19.1	83720.0	63623.8	33138.3
www.factoryschool.org	4h	2.1	2.1	0.0	0.0	0.0
real.scripps.com	1h	0.5	2.1	0.0	0.0	0.0
rslb.eonstreams.com	1h	0.0	0.0	0.0	0.0	0.0
LDNS: ispcom1						
mfile.akamai.com	2h	100.0	100.0	16.0	45.1	88.3
boss.streamos.com	10m	81.8	100.0	43.0	133.1	335.6
www.iuma.com	5m	23.1	100.0	450.0	980.7	1843.7
rslb.eonstreams.com	1h	24.6	70.2	5379.0	10646.7	11326.8
real.scripps.com	1h	19.4	51.1	9044.5	15077.7	17724.3
www.connectlive.com	20m	3.0	29.8	22876.0	32831.3	40797.7
www.factoryschool.org	4h	4.2	4.2	107115.0	107115.0	0.0
LDNS: isphome1						
boss.streamos.com	10m	50.7	100.0	258.0	583.9	939.6
mfile.akamai.com	2h	90.6	100.0	259.5	719.7	1140.1
www.iuma.com	5m	9.9	91.5	1295.0	2736.2	4227.6
real.scripps.com	1h	17.3	53.2	13346.0	17605.4	16333.3
rslb.eonstreams.com	1h	4.7	19.1	73699.0	78671.5	58160.3
www.connectlive.com	20m	1.4	14.9	66326.5	73452.0	70380.0
www.factoryschool.org	4h	4.2	4.2	310654.0	310654.0	0.0

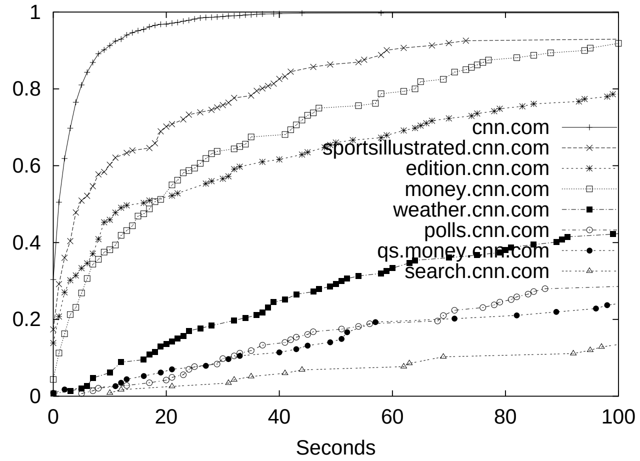

Figure 6: Cumulative Distribution Function for Measured Cache Gap of CNN Servers on ispcom1

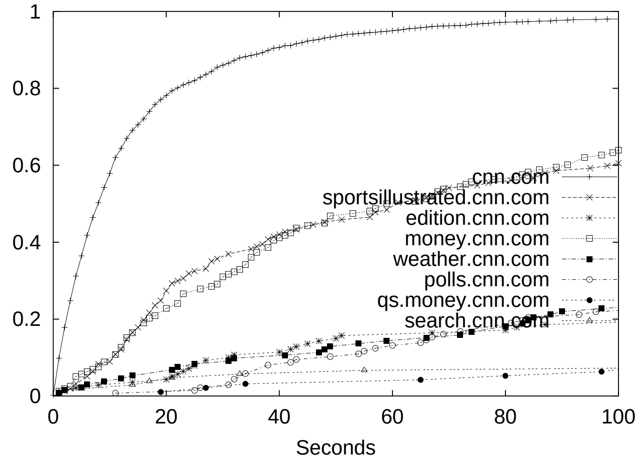

Figure 7: Cumulative Distribution Function for Measured Cache Gap of CNN Servers on isphome1

servers for study, although in the first three cases a DNS request for one of these servers could be for a browser to obtain HTML rather than streaming content. The remaining four servers do not have a home page and we expect their use to only be for serving streaming content.

1. www.connectlive.com, 20m.
2. www.iuma.com, 5m.
3. www.factoryschool.org, 4h.
4. boss.streamos.com, 10m.
5. mfile.akamai.com, 2h.
6. rslb.eonstreams.com, 1h.
7. real.scripps.com, 1h.

Results for one of the educational sites plus two of the ISPs are shown in Table 5. Because the cache hit rates are relatively lower than previous applications we do not show CDFs for the measured cache gaps. Note that we compare the servers using the hit rate for sampling every 4 hours as a common comparison interval.

The results show that the mfile.akamai.com and boss.streamos.com servers have the most frequent access across the three LDNSs. The www.factoryschool.org server has almost no lookups indicating that content on this server was rarely accessed by users of these LDNSs over the one week period of our study. These results can help to focus the content examined in characterization studies such as [11].

7.4 Network Games

Our next focus area was network games where we use the lookup of game servers by applications as an indication that games are being played. We studied usage of three game servers:

1. useast.battle.net, 12h.
2. uswest.battle.net, 1h.
3. master.gamespy.com, 1h.

The first two servers, located on each coast of the United States, are used to serve users of games such as WarCraft and StarCraft [1]. The last server is part of the GameSpy network, which is responsible for games such as Counter-Strike [8].

Results for these servers at selected LDNSs are shown in Table 6. Surprisingly the two battle.net servers use dramatically different ATTLs. Consequently the common comparison interval for all servers is 12 hours. The results do not show a clear distinction in relative popularity between the servers. The preference of useast.battle.net by edu5 and ispcom1 users confirms that both LDNSs are indeed located on the east coast while the geographic balance for isphome1 users indicates the users are more geographically dispersed.

7.5 Grid Computing

We examined relative frequency of use for two better known grid computing applications—SETI@home [19] and distributed.net [6]. These applications execute as low priority processes or as screen savers on machines and allow interested users to contribute CPU cycles to a computational grid. We studied three servers to which computational results are reported.

Table 6: Cache Hit % and Measured Cache Gap (sec) of Network Game Servers for Selected Local DNS Servers

Server	ATTL	Hit %	12H Hit %	Med	Mean	StDev
LDNS: edu5						
useast.battle.net	12h	92.9	92.9	921.5	3209.9	4867.3
uswest.battle.net	1h	14.5	76.9	14690.0	20347.9	23553.4
master.gamespy.com	1h	33.7	100.0	2316.0	6796.0	11792.3
LDNS: ispcom1						
useast.battle.net	12h	100.0	100.0	11.0	27.4	38.6
uswest.battle.net	1h	91.6	100.0	102.0	317.3	576.3
master.gamespy.com	1h	98.2	100.0	12.0	54.2	163.4
LDNS: isphome1						
useast.battle.net	12h	100.0	100.0	150.0	471.2	795.1
uswest.battle.net	1h	86.7	100.0	168.0	555.0	1488.1
master.gamespy.com	1h	87.3	100.0	150.5	531.5	1007.3

1. us.v27.distributed.net, 15m.
2. us.v29.distributed.net, 15m.
3. shserver2.ssl.berkeley.edu, 4h.

The first two are data servers for two different versions of the distributed.net application [7]. The other server is the data server used by SETI@home [10]. Table 7 shows results for these three servers for the users of three LDNSs.

The results show that SETI@home is generally more popular for these sets of users. These results, and others not shown, indicate that different versions of distributed.net are more popular at different LDNSs.

8. SUMMARY AND FUTURE WORK

This work introduces a novel methodology for inferring the usage patterns for Internet applications by a group of users. We use the information contained in the LDNS used by these applications to lookup server names with the DNS mechanism. While this methodology does not provide precise usage information as might be obtained from logs or network packet traces, it can be applied using the DNS mechanism without the difficulty of obtaining logs or traces from an organization. The methodology can be used to collect usage information from any LDNS that accepts DNS queries allowing usage patterns for a populations of users to be targeted based upon the LDNSs chosen to study.

We have described how the methodology works and shown that it can provide coarse usage frequency using cache hits as a metric as well as more precise usage information by using measured cache gaps to approximate request intervals for more frequently accessed servers. We show that the cache gaps work to approximate request intervals using a log of actual DNS requests and also identify situations where the methodology does not work well. Problems occur when a LDNS cache receives periodic requests for a server name, which are indicated by discontinuities in the CDF of the measured cache gaps for the name.

We go on to show how the methodology can be applied to a number of application domains where the DNS lookup of a particular server name implies information about the use of an application. Keying on the frequency that the server name appears in the LDNS cache, we can identify whether an application is being used and if so then to understand its relative popularity. We infer the relative popularity of servers used for Web searching, handling content requests at the CNN web site, repositories of streaming content, network games and grid computing.

The work raises a number of directions for future work. Clearly using the methodology for other application domains is of interest. Possible uses that we can immediately identify are to track the use of instant messaging (IM) servers to determine the relative popularity of different IM services. The use of peer-to-peer networks could be tracked if they use well-known "super nodes" that require a DNS lookup when new peers join. Many Content Distribution Networks (CDNs) use DNS to direct clients to different servers allowing the frequency of use for these servers to be tracked. The presence of DNS `MX` records in the cache can be used to track the popularity of mail servers. In general, the relative popularity of any Internet application that uses distinguished server names can be tracked. The methodology can also be used for longitudinal study on changes in request patterns.

Another direction for future work is to refine and better understand the methodology. We plan to do more extensive testing on the methodology with known LDNS request logs. We could also use known ADNS request logs to determine LDNSs in the Internet. We also plan to investigate whether it is possible to compensate for periodicity in the DNS requests so results exhibiting discontinuities in the cache gap CDFs can be used. We also plan to better explore the time-of-day effect and focus on "peak-time" performance, which may make sense for the LDNS of a single site, but may not for LDNSs serving clients in a range of time zones. Finally, we need to adapt the methodology if the ATTL of a server name changes.

A direction of work is to explore how other types of caches can be used to infer popularity. For example, Web caches contain objects that could be considered popular, although determining the content of the objects may be difficult unless the cache has a mechanism to reveal its contents. Normally responses to Web requests do not indicate if an object was served from a cache nor do they provide information on how long an object has resided in the cache. It may be possible to infer cache content based on retrieval latency if there is a big difference between latency of cached and non-cached retrievals as we did in [22] for determining the source of

Table 7: Cache Hit % and Measured Cache Gap (sec) of Grid Computing Servers for Selected Local DNS Servers

Server	ATTL	Hit %	4H Hit %	Med	Mean	StDev
LDNS: edu5						
us.v27.distributed.net	15m	9.1	82.9	7262.0	7856.4	6913.5
us.v29.distributed.net	15m	0.6	4.9	0.0	0.0	0.0
shserver2.ssl.berkeley.edu	4h	97.6	97.6	240.0	299.1	266.3
LDNS: ispcom1						
us.v27.distributed.net	15m	19.9	100.0	2806.0	3511.9	3027.2
us.v29.distributed.net	15m	25.0	92.7	2202.0	3866.9	5233.6
shserver2.ssl.berkeley.edu	4h	97.6	97.6	87.0	279.4	1098.6
LDNS: isphome1						
us.v27.distributed.net	15m	25.1	90.2	750.0	2496.5	4336.4
us.v29.distributed.net	15m	47.0	100.0	119.0	1336.8	3083.8
shserver2.ssl.berkeley.edu	4h	95.2	95.2	268.5	882.3	1947.2

non-authoritative DNS records.

A final direction that needs more exploration is the tradeoff in using caches to identify popular content while protecting privacy concerns for users. Given that any user can send a query to a LDNS cache, there is certainly the potential that a user could corroborate DNS cache contents with information on which users are active to make inferences about what these users are doing. While access control lists (ACLs) for LDNSs can prevent access to these LDNSs by "outside" users, ACLs cannot be used to prevent tracking of the LDNS cache by the legitimate users of the LDNS.

9. ACKNOWLEDGMENTS

We thank David Finkel and Jayson Wilbur for helping us to explore the relationship between request intervals and measured cache gaps. We also thank James Nichols for providing information on how servers are used by instant messaging clients. Finally, we thank the anonymous reviewers for their detailed and helpful comments on technical aspects of the paper and on the presentation.

10. REFERENCES

[1] Blizzard entertainment: Technical support site. `http://www.blizzard.com/support/?id=mgi0546p`.

[2] N. Brownlee, kc Claffy, and E. Nemeth. DNS measurements at a root server. In *Proceedings of the IEEE 2001 Global Telecommunications Conference*. IEEE, 2001.

[3] E. Cohen and H. Kaplan. Proactive caching of DNS records: Addressing a performance bottleneck. In *Proceedings of the Symposium on Applications and the Internet*, San Diego-Mission Valley, CA, USA, Jan. 2001. IEEE-TCI.

[4] C. D. Cranor, E. Gansner, B. Krishnamurthy, and O. Spatscheck. Characterizing large DNS traces using graphs. In *Proceedings of the ACM SIGCOMM Internet Measurement Workshop*, San Francisco, CA, Nov. 2001.

[5] P. Danzig, K. Obraczka, and A. Kumar. An analysis of wide-area name server traffic: A study of the Internet domain name system. In *Proceedings of the ACM SIGCOMM '92 Conference*, Baltimore, MD, Aug. 1992. ACM.

[6] distributed.net. `http://distributed.net/`.

[7] distributed.net faq-o-matic. `http://n0cgi.distributed.net/faq/cache/158.html`, `http://n0cgi.distributed.net/faq/index.cgi?_recurse=1&file=40`.

[8] Syntech america's army server setup paperwork. `http://syntechsoftware.com/ST07052002P.php`.

[9] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. DNS performance and the effectiveness of caching. *IEEE/ACM Transactions on Networking*, 10(5):589–603, Oct. 2002.

[10] J. Leyden. Security glitch with SETI@home screensaver. *The Register*, April 2003. `http://www.theregister.co.uk/content/55/30124.html`.

[11] M. Li, M. Claypool, R. Kinicki, and J. Nichols. Characteristics of Streaming Media Stored on the Internet. Technical Report WPI-CS-TR-03-18, CS Department, Worcester Polytechnic Institute, May 2003.

[12] M. Mikhailov and C. E. Wills. Evaluating a new approach to strong web cache consistency with snapshots of collected content. In *Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungary, May 2003.

[13] P. Mockapetris. Domain names—concepts and facilities, November 1987. RFC 1034.

[14] P. Mockapetris. Domain names—implementation and specification, November 1987. RFC 1035.

[15] Nielsen//netratings. `http://www.nielsen-netratings.com/`.

[16] OIT help desk: DNS and SMTP information about various internet service providers. `http://www.helpdesk.umd.edu/documents/1/1989/`.

[17] Top 100 web sites (you didn't know you couldn't live without). *PC Magazine*, March 2003.

[18] Ranking.com. `http://ranking.com/`.

[19] Seti@home, the search for extraterrestrial intelligence. `http://setiathome.ssl.berkeley.edu/`.

[20] A. Shaikh, R. Tewari, and M. Agrawal. On the

effectiveness of DNS-based server selection. In *Proceedings of the IEEE Infocom 2001 Conference*, Anchorage, Alaska USA, Apr. 2001.

[21] D. Sullivan. The major search engines and directories. *SearchEngineWatch.com*, April 2003.

[22] C. E. Wills and H. Shang. The contribution of DNS lookup costs to web object retrieval. Technical Report WPI-CS-TR-00-12, Computer Science Department, Worcester Polytechnic Institute, July 2000. `http://www.cs.wpi.edu/~cew/papers/tr00-12.ps.gz`.