An Analysis of Internet Content Delivery Systems

Stefan Saroiu, Krishna P. Gummadi, Richard J. Dunn, Steven D. Gribble, and Henry M. Levy
Department of Computer Science & Engineering
University of Washington
{tzoompy,gummadi,rdunn,gribble,levy}@cs.washington.edu

Abstract

In the span of only a few years, the Internet has experienced an astronomical increase in the use of specialized content delivery systems, such as content delivery networks and peer-to-peer file sharing systems. Therefore, an understanding of content delivery on the Internet now requires a detailed understanding of how these systems are used in practice.

This paper examines content delivery from the point of view of four content delivery systems: HTTP web traffic, the Akamai content delivery network, and Kazaa and Gnutella peer-to-peer file sharing traffic. We collected a trace of all incoming and outgoing network traffic at the University of Washington, a large university with over 60,000 students, faculty, and staff. From this trace, we isolated and characterized traffic belonging to each of these four delivery classes. Our results (1) quantify the rapidly increasing importance of new content delivery systems, particularly peer-to-peer networks, (2) characterize the behavior of these systems from the perspectives of clients, objects, and servers, and (3) derive implications for caching in these systems.

1 Introduction

Few things compare with the growth of the Internet over the last decade, except perhaps its growth in the last several years. A key challenge for Internet infrastructure has been delivering increasingly complex data to a voracious and growing user population. The need to scale has led to the development of thousand-node clusters, global-scale content delivery networks, and more recently, self-managing peer-to-peer structures. These content delivery mechanisms are rapidly changing the nature of Internet content delivery and traffic; therefore, an understanding of the modern Internet requires a detailed understanding of these new mechanisms and the data they serve.

This paper examines content delivery by focusing on four content delivery systems: HTTP web traffic, the Akamai content delivery network, and the Kazaa and Gnutella peer-to-peer file sharing systems. To perform the study, we traced all incoming and outgoing Internet traffic at the University of Washington, a large university with over 60,000 students, faculty, and staff. For this paper, we analyze a nine day trace that saw over 500 million transactions and over 20 terabytes of HTTP data. From this data, we provide a detailed characterization and comparison of content delivery systems, and in particular, the latest peer-to-peer workloads. Our results quantify: (1) the extent to which peer-to-peer traffic has overwhelmed web traffic as a leading consumer of Internet bandwidth, (2) the dramatic differences in the characteristics of objects being transferred as a result, (3) the impact of the two-way nature of peer-to-peer communication, and (4) the ways in which peer-to-peer systems are *not* scaling, despite their explicitly scalable design. For example, our measurements show that an average peer of the Kazaa peer-to-peer network consumes 90 times more bandwidth than an average web client in our environment. Overall, we present important implications for large organizations, service providers, network infrastructure, and general content delivery.

The paper is organized as follows. Section 2 presents an overview of the content delivery systems examined in this paper, as well as related work. Section 3 describes the measurement methodology we used to collect and process our data. In Section 4 we give a high-level overview of the workload we have traced at the University of Washington. Section 5 provides a detailed analysis of our trace from the perspective of objects, clients, and servers, focusing in particular on a comparison of peer-to-peer and web traffic. Section 6 evaluates the potential for caching in content delivery networks and peer-to-peer networks, and Section 7 concludes and summarizes our results.

2 Overview of Content Delivery Systems

Three dominant content delivery systems exist today: the client/server oriented world-wide web, content delivery networks, and peer-to-peer file sharing systems. At a high level, these systems serve the same role of distributing content to users. However, the architectures of these systems differ significantly, and the differences affect their performance, their workloads, and the role caching can play. In this section, we present the architectures of these systems and describe previous studies of their behavior.

2.1 The World-Wide Web (WWW)

The basic architecture of the web is simple: using the HTTP [16] protocol, web clients running on users' machines request objects from web servers. Previous studies have examined many aspects of the web, including web workloads [2, 8, 15, 29], characterizing web objects [3, 11], and even modeling the hyperlink structure of the web [6, 21]. These studies suggest that most web objects are small (5-10KB), but the distribution of object sizes is heavy-tailed and very large objects exist. Web objects are accessed with a Zipf popularity distribution, as are web servers. The number of web objects is enormous (in the billions) and rapidly growing; most web objects are static, but an increasing number are generated dynamically.

The HTTP protocol includes provisions for consistency management. HTTP headers include caching pragmas that affect whether or not an object may be cached, and if so, for how long. Web caching helps to alleviate load on servers and backbone links, and can also serve to decrease object access latencies. Much research has focused on Web proxy caching [4, 5, 7, 11, 12] and, more recently, on coordinating state among multiple, cooperating proxy caches [13, 30, 33]; some of these proposals aim to create global caching structures [27, 34]. The results of these studies generally indicate that cache hit rates of 40-50% are achievable, but that hit rate increases only logarithmically with client population [36] and is constrained by the increasing amount of dynamically generated and hence uncacheable content.

2.2 Content Delivery Networks (CDNs)

Content delivery networks are dedicated collections of servers located strategically across the wide-area Internet. Content providers, such as web sites or streaming video sources, contract with commercial CDNs to host and distribute content. CDNs are compelling to content providers because the responsibility for hosting content is offloaded to the CDN infrastructure. Once in a CDN, content is replicated across the wide area, and hence is highly available. Since most CDNs have servers in ISP points of presence, clients can access topologically nearby replicas with low latency. The largest CDNs have thousands of servers dispersed throughout the Internet and are capable of sustaining large workloads and traffic hot-spots.

CDNs are tightly integrated into the existing web architecture, relying either on DNS interposition [19, 32] or on URL rewriting at origin servers to redirect HTTP requests to the nearest CDN replica. As with the web, the unit of transfer in a CDN is an object, and objects are named by URLs. Unlike the web, content providers need not manage web servers, since clients' requests are redirected to replicas hosted by the CDN. In practice, CDNs typically host static content such as images, advertisements, or media clips; content providers manage their own dynamic content, although dynamically generated web pages might contain embedded objects served by the CDN.

Previous research has investigated the use and effectiveness of content delivery networks [14], although the proprietary and closed nature of these systems tends to impede investigation. Two recent studies [22, 23] confirm that CDNs reduce average download response times, but that DNS redirection techniques add noticeable overhead because of DNS latencies. In another study [18], the authors argue that the true benefit of CDNs is that they help clients avoid the worst-case of badly performing replicas, rather than routing clients to a truly optimal replica. To the best of our knowledge, no study has yet compared the workloads of CDNs with other content delivery architectures.

2.3 Peer-to-Peer Systems (P2P)

Peer-to-peer file sharing systems have surged in popularity in recent years. In a P2P system, peers collaborate to form a distributed system for the purpose of exchanging content. Peers that connect to the system typically behave as servers as well as clients: a file that one peer downloads is often made available for upload to other peers. Participation is purely voluntary, and a recent study [31] has shown that most content-serving hosts are run by end-users, suffer from low availability, and have relatively low capacity network connections (modem, cable modems, or DSL routers).

Users interact with a P2P system in two ways: they attempt to locate objects of interest by issuing search queries, and once relevant objects have been located, users issue download requests for the content. Unlike the web and CDN systems, the primary mode of usage for P2P systems is a non-interactive, batch-style download of content.

P2P systems differ in how they provide search capabilities to clients [37]. Some systems, such as Napster [28], have large, logically centralized indexes maintained by a single company; peers automatically upload lists of available files to the central index, and queries are answered using this index. Other systems, such as Gnutella [10] and Freenet [9], broadcast search requests over an overlay network connecting the peers. More recent P2P systems, including Kazaa [20], use a hybrid architecture in which some peers are elected as "supernodes" in order to index content available at peers in a nearby neighborhood.

P2P systems also differ in how downloads proceed, once an object of interest has been located. Most systems transfer content over a direct connection between the object provider and the peer that issued the download request. A latency-improving optimization in some systems is to download multiple object fragments in parallel from multiple replicas. A recent study [24] has found the peer-to-peer traffic of a small ISP to be highly repetitive, showing great potential for caching.

3 Methodology

We use *passive network monitoring* to collect traces of traffic flowing between the University of Washington (UW) and the rest of the Internet. UW connects to its ISPs via two border routers; one router handles outbound traffic and the other inbound traffic. These two routers are fully connected to four switches on each of the four campus backbones. Each switch has a monitoring port that is used to send copies of the incoming and outgoing packets to our monitoring host.

Our tracing infrastructure is based on software developed by Wolman and Voelker for previous studies [35, 36]. We added several new components to identify, capture, and analyze Kazaa and Gnutella peer-to-peer traffic and Akamai CDN traffic. Overall, the tracing and analysis software is approximately 26,000 lines of code. Our monitoring host is a dual-processor Dell Precision Workstation 530 with 2.0 GHz Pentium III Xeon CPUs, a Gigabit Ethernet SysKonnect SK-9843 network card, and running FreeBSD 4.5.

Our software installs a kernel packet filter [26] to deliver TCP packets to a user-level process. This process reconstructs TCP flows, identifies HTTP requests within the flows (properly handling persistent HTTP connections), and extracts HTTP headers and other metadata from the flows. Because Kazaa and Gnutella use HTTP to exchange files, this infrastructure is able to capture P2P downloads as well as WWW and Akamai traffic. We anonymize sensitive information such as IP addresses and URLs, and log all extracted data to disk in a compressed binary representation.

3.1 Distinguishing Traffic Types

Our trace captures two types of traffic: *HTTP traffic*, which can be further broken down into WWW, Akamai, Kazaa, and Gnutella transfers, and *non-HTTP TCP traffic*, including Kazaa and Gnutella search traffic. If an HTTP request is directed to port 80, 8080, or 443 (SSL), we classify both the request and the associated response as WWW traffic. Similarly, we use ports 6346 and 6347 to identify Gnutella HTTP traffic, and port 1214 to identify Kazaa HTTP traffic. A small part of our captured HTTP traffic remains unidentifiable; we believe that most of this traffic can be attributed to less popular peer-to-peer systems (e.g., Napster [28]) and by compromised hosts turned into IRC or web servers on ports other than 80, 8080, or 444. For non-HTTP traffic, we use the same Gnutella and Kazaa ports to identify P2P search traffic.

Some WWW traffic is served by the Akamai content delivery network [1]. Akamai has deployed over 13,000 servers in more than 1,000 networks around the world [25]. We identify Akamai traffic as any HTTP traffic served by any Akamai server. To obtain a list of Akamai servers, we collected a list of 25,318 unique authoritative name servers, and sent a recursive DNS query to each server for a name in an Akamai-managed domain (e.g., `a388.g.akamaitech.net`). Because Akamai redirects DNS queries to nearby Akamai servers, we were able to collect a list of 3,966 unique Akamai servers in 928 different networks.

For the remainder of this paper, we will use the following definitions when classifying traffic:

- **Akamai:** HTTP traffic on port 80, 8080, or 443 that is served by an Akamai server.
- **WWW:** HTTP traffic on port 80, 8080, or 443 that is not served by an Akamai server; thus, for all of the analysis within this paper, "WWW traffic" does not include Akamai traffic.
- **Gnutella:** HTTP traffic sent to ports 6346 or 6347 (this includes file transfers, but excludes search and control traffic).
- **Kazaa:** HTTP traffic sent to port 1214 (this includes file transfers, but excludes search and control traffic).
- **P2P:** the union of Gnutella and Kazaa.
- **non-HTTP TCP traffic:** any other TCP traffic, including protocols such as NNTP and SMTP, HTTP traffic to ports other than those listed above, traffic from other P2P systems, and control or search traffic on Gnutella and Kazaa.

3.2 The Traceability of P2P Traffic

Gnutella is an overlay network over which search requests are flooded. Peers issuing search requests receive a list of other peers that have matching content. From this list, the peer that issued the request initiates a direct connection with one of the matching peers to download content. Because the Gnutella overlay is not structured to be efficient with respect to the physical network topology, most downloads initiated by UW peers connect to external hosts, and are therefore captured in our traces.

Although the details of Kazaa's architecture are proprietary, some elements are known. The Kazaa network is a two-level overlay: some well-connected peers serving as "supernodes" build indexes of the content stored on nearby "regular" peers. To find content, regular peers issue search requests to their supernodes. Supernodes appear to communicate amongst themselves to satisfy queries, returning locations of matching objects to the requesting peer. Kazaa appears to direct peers to nearby objects, although the details of how this is done, or how successful the system is at doing it, are not known.

To download an object, a peer initiates one or more connections to other peers that have replicas of the object. The downloading peer may transfer the entire object in one connection from a single peer, or it may choose to download multiple fragments in parallel from multiple peers.

	WWW		Akamai		Kazaa		Gnutella	
	inbound	outbound	inbound	outbound	inbound	outbound	inbound	outbound
HTTP transactions	329,072,253	73,001,891	33,486,508	N/A	11,140,861	19,190,902	1,576,048	1,321,999
unique objects	72,818,997	3,412,647	1,558,852	N/A	111,437	166,442	5,274	2,092
clients	39,285	1,231,308	34,801	N/A	4,644	611,005	2,151	25,336
servers	403,087	9,821	350	N/A	281,026	3,888	20,582	412
bytes transferred	1.51 TB	3.02 TB	64.79 GB	N/A	1.78 TB	13.57 TB	28.76 GB	60.38 GB
median object size	1,976 B	4,646 B	2,001 B	N/A	3.75 MB	3.67 MB	4.26 MB	4.08 MB
mean object size	24,687 B	82,385 B	12,936 B	N/A	27.78 MB	19.07 MB	19.16 MB	9.78 MB

Table 1. HTTP trace summary statistics: trace statistics, broken down by content delivery system; *inbound* refers to transfers from Internet servers to UW clients, and *outbound* refers to transfers from UW servers to Internet clients. Our trace was collected over a nine day period, from Tuesday May 28th through Thursday June 6th, 2002.

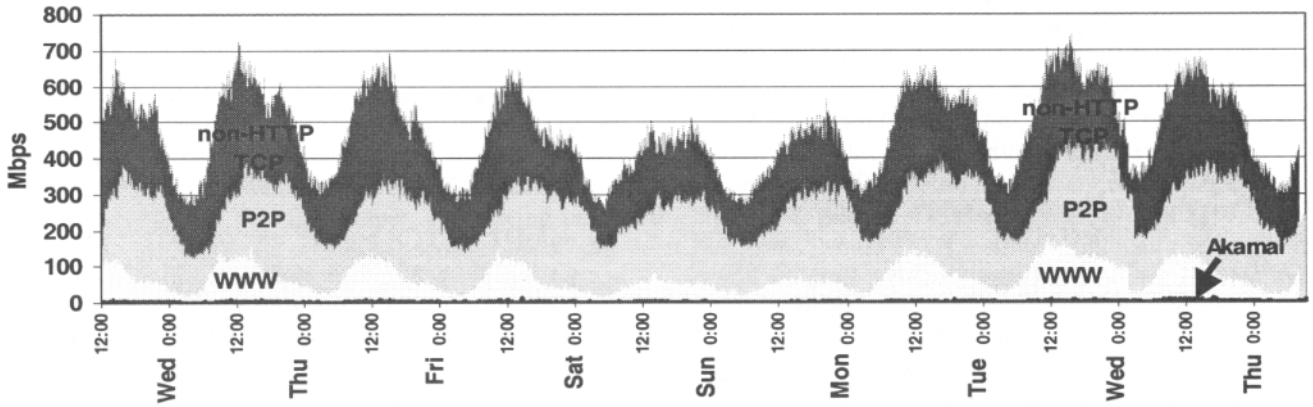

Figure 1. TCP bandwidth: total TCP bandwidth consumed by HTTP transfers for different content delivery systems. Each band is cumulative; this means that at noon on the first Wednesday, Akamai consumed approximately 10 Mbps, WWW consumed approximately 100 Mbps, P2P consumed approximately 200 Mbps, and non-HTTP TCP consumed approximately 300 Mbps, for a total of 610 Mbps.

The ability for a Kazaa peer to download an object in fragments complicates our trace. Download requests from external peers seen in our trace are often for fragments rather than entire objects.

4 High-Level Data Characteristics

This section presents a high-level characterization of our trace data. Table 1 shows summary statistics of object transfers. This table separates statistics from the four content delivery systems, and further separates inbound data (data requested by UW clients from outside servers) from outbound data (data requested by external clients from UW servers).

Despite its large client population, the University is a net *provider* rather than consumer of HTTP data, exporting 16.65 TB but importing only 3.44 TB. The peer-to-peer systems, and Kazaa in particular, account for a large percentage of the bytes exported and the total bytes transferred, despite their much smaller internal and external client populations. Much of this is attributable to a large difference in average object sizes between WWW and P2P systems.

The number of clients and servers in Table 1 shows the extent of participation in these systems. For the web, 39,285 UW clients accessed 403,437 Internet web servers, while for Kazaa, 4,644 UW clients accessed 281,026 external Internet servers. For Akamai, 34,801 UW clients download Akamai-hosted content provided by 350 different Akamai servers. In the reverse direction, 1,231,308 Internet clients accessed UW web content, while 611,005 clients accessed UW-hosted Kazaa content.

Figure 1 shows the total TCP bandwidth consumed in both directions over the trace period. The shaded areas show HTTP traffic, broken down by content delivery system; Kazaa and Gnutella traffic are grouped together under the label "P2P." All systems show a typical diurnal cycle. The smallest bandwidth consumer is Akamai, which currently constitutes only 0.2% of observed TCP traffic. Gnutella consumes 6.04%, and WWW traffic is the next largest, consuming 14.3% of TCP traffic. Kazaa is currently the largest contributor, consuming 36.9% of TCP bytes. These four content delivery systems account for 57% of total TCP traffic, leaving 43% for other TCP-based network protocols (streaming media, news, mail, and so on). TCP traffic represents over 97% of all network traffic at UW. This closely matches published data on Internet 2 usage [17].

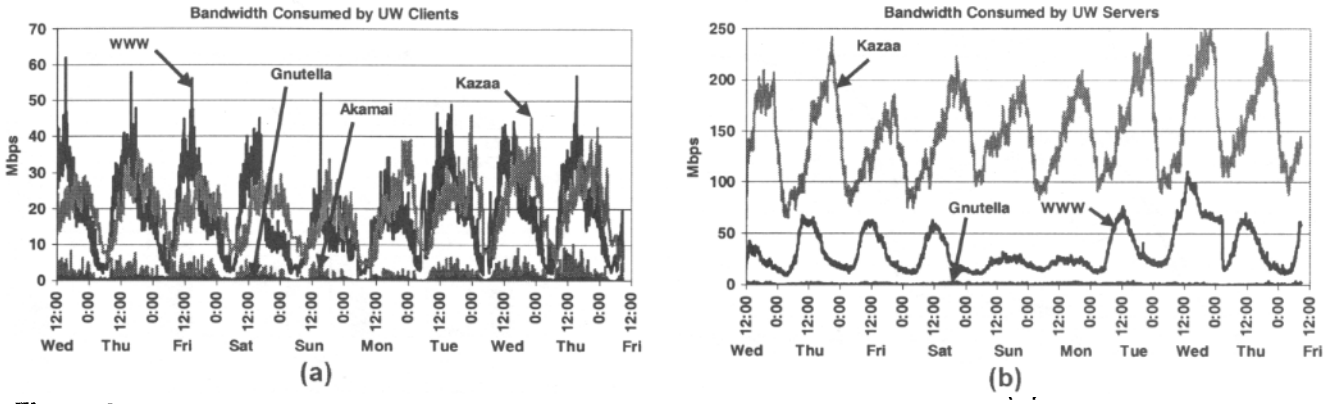

Figure 2. UW client and server TCP bandwidth: bandwidth over time (a) accountable to web and P2P downloads from UW clients, and (b) accountable to web and P2P uploads from UW servers.

Figures 2a and 2b show inbound and outbound data bandwidths, respectively. From Figure 2a we see that while both WWW and Kazaa have diurnal cycles, the cycles are offset in time, with WWW peaking in the middle of the day and Kazaa peaking late at night. For UW-initiated requests, WWW and Kazaa peak bandwidths have the same order of magnitude; however, for requests from external clients to UW servers, the peak Kazaa bandwidth dominates WWW by a factor of three. Note that the Y-axis scales of the graphs are different; WWW peak bandwidth is approximately the same in both directions, while external Kazaa clients consume 7.6 times more bandwidth than UW Kazaa clients.

Figure 3a and 3b show the top 10 content types requested by UW clients, ordered by bytes downloaded and number of downloads. While GIF and JPEG images account for 42% of requests, they account for only 16.3% of the bytes transferred. On the other hand, AVI and MPG videos, which account for 29.3% of the bytes transferred, constitute only 0.41% of requests. HTML is significant, accounting for 14.6% of bytes and 17.8% of requests. The 9.9% of bytes labelled "HASHED" in Figure 3a are Kazaa transfers that cannot be identified; of the non-hashed Kazaa traffic that can be identified, AVI and MPG account for 79% of the bytes, while 13.6% of the bytes are MP3.

It is interesting to compare these figures with corresponding measurements from our 1999 study of the same population [35]. Looking at bytes transferred as a percent of total HTTP traffic, HTML traffic has decreased 43% and GIF/JPG has decreased 59%. At the same time, AVI/MPG (and Quicktime) traffic has increased by nearly 400%, while MP3 traffic has increased by nearly 300%. (These percentages numbers include an estimate of the appropriate portion of the hashed bytes contributing to all content types).

In summary, this high-level characterization reveals substantial changes in content delivery systems usage in the Internet, as seen from the vantage point of UW. First, the balance of HTTP traffic has changed dramatically over the last several years, with P2P traffic overtaking WWW traffic as the largest contributor to HTTP bytes transferred. Second, although UW is a large publisher of web documents,

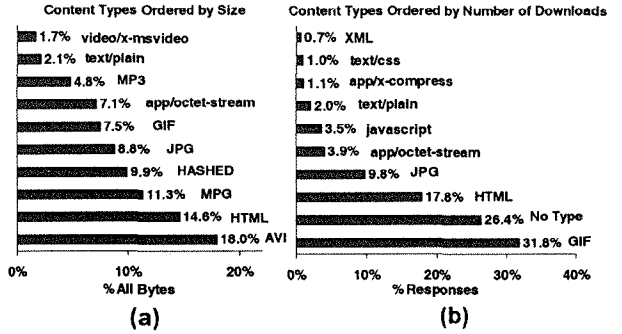

Figure 3. Content types downloaded by UW clients: a histogram of the top 10 content types downloaded by UW clients, across all four systems, ordered by (a) size and (b) number of downloads.

P2P traffic makes the University an even larger exporter of data. Finally, the mixture of object types downloaded by UW clients has changed, with video and audio accounting for a substantially larger fraction of traffic than three years ago, despite the small number of requests involving those data types.

5 Detailed Content Delivery Characteristics

The changes in Internet workload that we have observed raise several questions, including: (1) what are the properties of the new objects being delivered, (2) how are clients using the new content-delivery mechanisms, and (3) how do servers for new delivery services differ from those for the web? We attempt to answer these questions in the subsections below.

5.1 Objects

Data in Section 4 suggests that there is a substantial difference in typical object size between P2P and WWW traffic. Figure 4 illustrates this in dramatic detail. Not surprisingly, Akamai and WWW object sizes track each other fairly closely. The median WWW object is approximately 2KB, which matches previous measurement studies [15]. The Kazaa and Gnutella curves are strikingly different from the WWW; the median object size for these P2P systems is

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.