YouTube Traffic Characterization: A View From the Edge

Phillipa Gill[¶] Martin Arlitt[‡¶] Zongpeng Li[¶] Anirban Mahanti[§]
¶Department of Computer Science, University of Calgary, Canada
‡Enterprise Systems & Software Lab, HP Labs, Palo Alto, USA
§Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, India

ABSTRACT

This paper presents a traffic characterization study of the popular video sharing service, YouTube. Over a three month period we observed almost 25 million transactions between users on an edge network and YouTube, including more than 600,000 video downloads. We also monitored the globally popular videos over this period of time.

In the paper we examine usage patterns, file properties, popularity and referencing characteristics, and transfer behaviors of YouTube, and compare them to traditional Web and media streaming workload characteristics. We conclude the paper with a discussion of the implications of the observed characteristics. For example, we find that as with the traditional Web, caching could improve the end user experience, reduce network bandwidth consumption, and reduce the load on YouTube's core server infrastructure. Unlike traditional Web caching, Web 2.0 provides additional metadata that should be exploited to improve the effectiveness of strategies like caching.

Categories and Subject Descriptors

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

General Terms

Measurement, Performance

Keywords

YouTube, Web 2.0, Multimedia, Characterization

1. INTRODUCTION

The Web is slowly but steadily undergoing a metamorphosis as more and more users are able to create, share, and distribute content on the Web. This shift toward "user generated" content represents one of the biggest changes of the Web since its inception in the early 1990's. This paradigm shift has resulted in a surge in popularity of Web sites that enable users to build social networks and share content. Today, user generated content available on the Web includes textual information contained in Weblogs (blogs) [41], photos on sites such as Flickr [22] and Facebook [21], and videos on sites such as FlixHunt [23] and YouTube [42]. Collectively, these types of Web sites are referred to in the media as Web 2.0 (to distinguish these from the so-called Web 1.0 sites that host content from established providers) [12].

Web 2.0 changes how users participate in the Web. Instead of consuming content posted by a single administrator, users are now able to post their own content and view content posted by their peers. Some Web 2.0 sites, for example MySpace [35] and Facebook [21], promulgate social networking by allowing individuals with similar interests to form social groups. Tagging [5, 37], a feature that allows users to associate words or phrases ("tags") with content they post or view on a Web page, is extensively used on some Web 2.0 sites to categorize and organize content [22].

Adoption of Web 2.0 has been widespread, with users of all ages participating in posting as well as viewing content [33]. The diversity of participants in Web 2.0 is possible because of the low barrier to entry into these online communities. Many Web 2.0 sites are designed such that signing up and posting content are relatively easy. This enables users who may not be technically savvy to participate alongside more experienced users.

As the popularity of Web 2.0 sites grows, and as the availability of consumer broadband increases, the sheer volume of data exchanged for Web 2.0 traffic has the potential to severely strain the resources of both centralized servers and edge networks serving Web 2.0 users. Understanding Web 2.0 workloads will aid in network management, capacity planning, and the design of new systems. While there are extensive studies of traditional Web workload [6, 7, 17, 31], there have been no substantive studies of Web 2.0 workloads in the literature. Our work aims to fill this gap and strives to provide insights into how user generated content is viewed and distributed on the Internet.

In this paper, we analyze and characterize one such Web 2.0 site, YouTube [42], the largest video sharing site on the Internet [29]. According to estimates, with 100 million video views per day YouTube accounts for approximately 60% of the videos watched on the Internet; YouTube is also growing at a rapid pace, with 65,000 video uploads per day [40]. This constant growth of YouTube makes capturing its behavior by examining a single point in time almost impossible. Our analysis is based on three months of data that reflects trends of YouTube traffic from both a local campus network and a global (i.e., Internet wide) perspective. Locally, we consider the network resources consumed by YouTube traffic as

IMC'07, October 24-26, 2007, San Diego, California, USA.

well as the viewing habits of campus YouTube users. Globally, we consider characteristics of the most popular videos on YouTube and examine the relationship between globally popular videos and videos that are popular on campus.

The main contributions of our paper are threefold. First, we introduce an efficient measurement framework that enables us to monitor a popular and resource intensive Web 2.0 application over an extended period of time (while protecting user privacy). Second, we provide one of the first extensive characterization studies of Web 2.0 traffic. Third, we examine the implications of the observed characteristics. In particular, we analyze a wide range of features of YouTube traffic, including usage patterns, file properties, popularity and referencing behaviors, and transfer characteristics, which we compare to characteristics of traditional Web and streaming media workloads. For example, we observe that a small fraction of the requests to YouTube are for videos, but video downloads account for almost all of the bytes transferred, as video file sizes are orders of magnitude larger than files of other content types. Although similar properties have been observed for other Internet applications, with Web 2.0 the impact may be more significant, as it is for content that appeals to a much larger audience. An obvious performance and scalability enhancement is to utilize caching effectively. Although caching has been thoroughly studied for "traditional" Web workloads, there are differences to consider for Web 2.0. For example, the ability for anyone to create content and make it available online implies there will be sustained supply of content. This can reduce the effectiveness of caching; indeed, we observe a lower concentration of references than has been observed in traditional Web workloads. However, Web 2.0 provides an abundance of meta-data (compared to the traditional Web; e.g., user ratings, video categories, etc.); this meta-data can and should be exploited by Web 2.0 caching, in order to be more effective.

The remainder of the paper is structured as follows. Section 2 presents background information on YouTube. We discuss related work in Section 3. Our data collection framework is described in Section 4, followed by a high-level analysis of the collected data in Section 5. The next four sections characterize the YouTube workload in more detail. Section 6 characterizes YouTube's video and non-video files. The popularity characteristics of video files accessed by users on our campus is analyzed in Section 7, and locality properties of our campus YouTube traffic is analyzed in Section 8. In Section 9, we characterize the transfer size and durations for YouTube traffic on our campus network. Section 10 describes the implications of the workload characteristics we identified. We conclude the paper in Section 11 with a summary of our contributions and a discussion of future work.

2. BACKGROUND

YouTube was founded in February 2005 as a Web site that enables users to easily share video content. As YouTube expanded, features were added to facilitate social networking among its users. Users can "tag" their uploaded videos with keywords or phrases that best describe their content, and these tags are used by YouTube to provide users with a list of related videos. Tagging, social networking, and the abundance of user generated content make YouTube the quintessential Web 2.0 site. According to recent media reports, YouTube is the largest video sharing Web site on the Internet with over 100 million video accesses per day and 65,000 video uploads per day [40]. Time magazine's 2006 year end issue named "You" as the person of the year, as an homage to YouTube and other Web 2.0 users. Due to the incredible popularity of YouTube, it attracted the attention of numerous investors. In November 2006, YouTube was acquired by Google for $1.65 billion US.

One of the keys to YouTube's success is its use of Adobe's Flash Video (FLV) format for video delivery. While users may upload content in a variety of media formats (e.g., WMV, MPEG and AVI), YouTube converts them to Flash Video before posting them. This enables users to watch the videos without downloading any additional browser plugins provided they have the Flash Player 7 installed. It is estimated that over 90% of clients have Flash Player 7 installed.[1] To enable playback of the flash video before the content is completely downloaded, YouTube relies on Adobe's *progressive download* technology.

Traditional download-and-play requires the full FLV file to be downloaded before playback can begin. Adobe's progressive download feature allows the playback to begin without downloading the entire file. This is accomplished using ActionScript commands that supply the FLV file to the player as it is being downloaded, enabling playback of the partially downloaded file. Progressive download works with Web servers and video content is delivered using HTTP/TCP. This delivery technique is sometimes referred to as *pseudo streaming* to distinguish it from traditional media streaming. Traditional on-demand streaming of stored media files typically requires the use of dedicated streaming servers that facilitate client-server interaction during the course of the video playback. This interaction may be used for adaptation of video quality or user interactions such as fast forward or rewind operations.

While video content is usually the focus of a visit to the YouTube Web site, there are many file transfers that happen behind the scenes to embed the video file and display the surrounding Web site content. For example, when a user clicks on a video of interest, a GET request for the title HTML page for the requested video is made. This HTML page typically includes references to a number of Javascript files. These scripts are responsible for embedding the Shockwave Flash (SWF) player file, and other peripheral tasks such as processing video ratings and comments. The SWF file is relatively small (26 KB), so the page loads quickly. Once the player is embedded, a request for the FLV video file is issued. The FLV video file is downloaded to the user's computer using an HTTP GET request, which is serviced by either a YouTube server or a server from a content distribution network (CDN).

3. RELATED WORK

There are numerous studies of "traditional" Web (now referred to as Web 1.0) workloads. Cunha *et al.* characterized a set of Web browser traces [18], while Gribble and Brewer analyzed HTTP traces from a dial-in modem pool [24]. Both examine characteristics such as access patterns, object types, and object sizes. Arlitt and Williamson identified a set of ten characteristics common to Web server workloads [7]. Arlitt and Jin examined the much busier

[1] http://www.adobe.com/products/player_census/flashplayer/version_penetration.html

World Cup 1998 Web site, and verified that these characteristics existed [6]. Mahanti *et al.* [31] and Duska *et al.* [20] characterized Web proxy workloads. A common conclusion from all of these studies was that caching had the potential to improve both the user experience (i.e., through reduced latency) and the scalability of the Web (i.e., by distributing the workload). Our work is complementary, in that it examines a Web 2.0 workload for similar characteristics and opportunities for infrastructure improvements.

Characterization of both stored and live media streaming has also received considerable attention in the literature. Characteristics of media files on the Web have been studied using a crawling or searching perspective, originating from an edge network [2, 28], or by analyzing traces collected in the network [15, 25].

In 1998, Acharya *et al.* presented one of the earliest known study of the characteristics of streaming media files stored on Web servers [2]. This was followed by Chesire *et al.* [15] who analyzed a week-long trace, collected in 2000 from their campus's Internet gateway, of live and on-demand RTSP sessions. They found that most media streams viewed on their campus were encoded at low bit rates suitable for streaming to dial-up users, were typically less than 1 MB in size, and had durations less than 10 minutes. In addition, they found media file popularity to be Zipf-like.

In 2003, Li *et al.* [28] crawled 17 million Web pages for stored audio/video files and discovered 30,000 such files. Analyzing these files, they reported several observations: media durations are long-tailed; media files are typically encoded in proprietary formats; and most video files are encoded at bit rates targeted at broadband users.

In 2004, Sripanidkulchai *et al.* [38] analyzed a workload of live media streams collected from a large CDN. They observe that media popularity follows a 2-mode Zipf distribution. They also observe exponentially distributed client arrival times within small time windows and heavy-tailed session durations.

Workloads from media servers in corporate, university, and commercial environments environments have also been studied [3, 4, 14, 16, 27, 43]. For example, Almeida *et al.* [4] presented a detailed analysis of workloads from two media servers (eTeach and BIBS) located at two large universities in the United States. They found file popularity can be modeled as a concatenation of two Zipf distributions, and that client interarrival times followed the exponential distribution in the case of eTeach and the Pareto distribution in the case of BIBS. They also observed uniform access to all segments of popular files whereas access to segments of infrequently accessed files was non-uniform. The authors also observed a lack of temporal locality in client requests.

Cherkasova and Gupta [14] analyzed the workloads of two corporate media servers. They report that video popularity is Zipf-like, that a significant fraction of the total requests and bytes transferred were for new content, and that most accesses to a file occurred soon after the files were made available on the servers.

More recently, Yu *et al.* [43] presented an analysis of the file reference characteristics and the user behavior of the Powerinfo system, a production video-on-demand system deployed in major Chinese cities by China Telecom. The system mostly hosts older television programs encoded in MPEG format. The authors analyzed 217 days of access logs from one city with 150,000 users. Their access logs recorded 6,700 unique video requests and a total of 21 million video requests. They found that: request arrival rate is strongly influenced by time of day, request arrivals can be modeled by a modified Poisson distribution, video popularity follows the Zipf distribution, and user interest in videos is fueled by several factors such as the list of videos on the most recommended list and the availability of new videos.

Recently, we have discovered parallel studies of YouTube [13, 26]. Both of these studies employ crawling for characterizing YouTube video files. Our work is complementary to these aforementioned works, with a distinguishing factor being our measurement based approach to characterizing usage of YouTube from an edge network perspective.

4. DATA COLLECTION FRAMEWORK

YouTube's workload is a moving target. Everyday, new videos are added, new ratings are submitted, and new comments are posted. The popularity of videos also changes on a daily basis. In this paper, we propose a multilevel approach to capturing YouTube traffic and understanding its workload characteristics. First, we monitor YouTube usage on our local (University of Calgary) campus network. Our campus consists of approximately 28,000 students and 5,300 faculty and staff [1]. By considering local YouTube usage we are able to understand how YouTube may be used by clients of other large edge networks. Section 4.1 describes our local data collection methodology. Second, we collect statistics on the most popular videos on the YouTube site. Section 4.2 explains our global data collection methodology. By keeping statistics of both local and global YouTube usage we are able to compare and contrast characteristics of videos that are popular at both the local and global level.

4.1 Data Collection of Edge YouTube Usage

An enabling step in this work was the collection of data from an edge network. Our goals in data collection were to:

- collect data on all YouTube usage at the University of Calgary network
- gather such data for an extended period of time
- protect user privacy

This conceptually simple task proved challenging, for a number of reasons. One challenge is the global popularity of YouTube. Due to its popularity, YouTube's delivery infrastructure is comprised of many servers, including some from (one or more) Content Distribution Networks (CDNs). A second challenge is our network monitor has limited CPU and storage resources,[2] thus making storage of lengthy full packet traces infeasible. A third challenge is our campus recently upgraded from a 100 to a 300 Mb/s full-duplex network link to the Internet; users on campus were happy to increase their Internet usage, which places greater pressure on our aging network monitor. Figure 1 shows the aggregate bandwidth (inbound + outbound) consumed on our campus Internet link during the collection period.

The data collection methodology we used to address these challenges is as follows:

[2]Our monitor was purchased in spring 2003, when our Internet connection was only 12 Mb/s. Our monitor has two Intel Pentium III 1.4 GHz processors, 2 GB RAM, and two 70 GB drives.

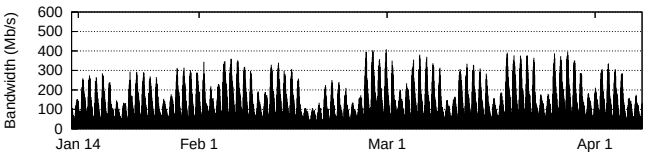

Figure 1: Aggregate Campus Internet Bandwidth During Collection Period

- identify a set of servers that provide YouTube content
- use `bro` [9] to collect summary information on each HTTP transaction involving one of those servers
- restart bro daily, compress the previous day's log

We identified the servers to monitor a priori. Initially we used `tcpdump` [39] to gather traces on a workstation while we browsed the YouTube site. This provided a sample of the servers used to deliver YouTube content. We then used `whois` to determine the networks that the server's IP addresses were affiliated with. We identified two networks (youtube and youtube2) that were assigned to YouTube. For our long-term data collection, we gather all HTTP transactions involving any IP address on these two networks. We also identified one CDN (Limelight Networks) delivering YouTube content. Extracting traffic for this CDN required a slightly different approach, as the Limelight CDN also serves traffic for other popular Web 2.0 sites such as Facebook and MySpace. Fortunately, Limelight incorporates YouTube into the fully qualified domain name (FQDN) of each server, so we were able to consider only the transactions including an HTTP `Host:` field that included the term "youtube".[3]

We used `bro` to extract summaries of each YouTube HTTP transaction in real-time. We chose `bro` because it implements many of the functions we require; we just had to write a script to handle events of interest. For each transaction we record a variety of data about the TCP connection (e.g., duration, initial RTT, start and end sequence numbers), the HTTP request (e.g., the method, URL, `Host:` name), the HTTP response (e.g., status code, content length, date). In this study, application level characteristics are our primary interest. As a result, our analysis focuses on the HTTP data that we collect.

To protect user privacy, we convert the YouTube visitor identifier that is collected from the HTTP header into a unique integer. Furthermore, the mapping is not recorded to disk, and the mapping is only valid for a 24 hour period (i.e., until `bro` is restarted). This prevents us from analyzing some aspects of user longevity, but protects user privacy.

After initial experimentation with `bro` on our monitor, we found it necessary to add an additional field to each transaction summary. This field indicates the parsing status of each transaction, which falls into one of four categories: **Complete**, the entire transaction was successfully parsed; **Interrupted**, the TCP connection was reset before the transaction was complete; **Gap**, the monitor missed a packet, and thus `bro` was unable to parse the remainder of the transaction; **Failure**, `bro` was unable to parse the transaction for an unknown reason.

[3]Unfortunately, we still had to process traffic from other sites on the Limelight network, as multiple FQDNs often mapped to the same IP address.

Table 1: Breakdown of Transactions

Category	Transactions	% of Total
Completed	22,403,657	90.82
Interrupted	462,903	1.88
Gapped	383,878	1.56
Failed	1,418,178	5.75
Total	24,668,616	100.01

Table 2: Breakdown of Video Transactions

Category	Transactions	% of Total
Completed	154,294	24.66
Interrupted	151,687	24.25
Gapped	319,612	51.09
Total	625,593	100.00

Table 1 summarizes the prevalence of each of the transaction categories.[4] As we would expect, most transactions have a "Complete" status. About 6% of transactions "failed". For transactions in this category we have no information from HTTP headers. The two most likely reasons for failed transactions are: our monitor dropped a packet in the connection before the HTTP headers were parsed; or the TCP connection was not established in an expected manner, so our script did not know how to handle it properly.[5] Unfortunately, as we summarize each transaction in real-time and do not retain the raw packet traces, we do not have any definitive evidence to determine the prevalence of each. However, neither of these issues are related to the type of object being transferred, so it is unlikely that a disproportionate fraction of failed transactions were for video objects. We record the number of transactions that ended up in this category to ensure we are gathering information on the majority of identified YouTube transactions. Our analyses in the remainder of this paper ignores the failed transactions.

The breakdown of transactions for video requests is shown in Table 2. For video requests, only about one quarter of the transactions were complete. The main reason for this is the large number of transactions with a gap. As YouTube traffic increased on our campus, we observed that during busy periods our monitor (when running `bro`) could not keep up with the network load.[6] This resulted in some transaction summaries being incomplete due to "gaps" in a TCP connection's sequence number space. As Table 2 indicates, most of the gaps occur in video transactions. This happens because the video transactions achieve much higher download rates than most other (smaller) transactions, thus placing a higher load on our monitor. It is important to note, however, that most of the data we use is from the HTTP headers, and these are seen in the first few packets exchanged in a transaction, when the transfer rates are lower. As a result, we are still able to apply all of our analyses to transactions in this category, except those analyses which require the "Transfer Duration".

Approximately 24% of video transactions fall into the "interrupted" category. We can also use these transactions in most analyses, as the interruptions occur after the exchange of HTTP headers. We argue that there are two primary reasons why a video download may fall into this category:

[4]The total is 100.01% due to rounding error.

[5]Our script expects a three packet establishment handshake for each TCP connection: SYN, SYN ACK, ACK. If a client's TCP stack behaves differently from this, our script will mark the transaction as failed.

[6]In the near future we plan to upgrade to a more powerful monitor.

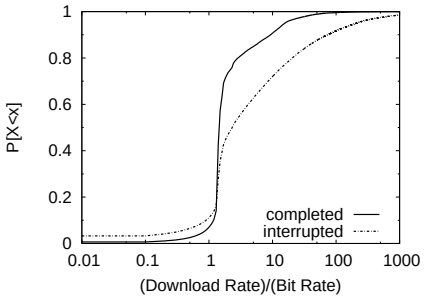

Figure 2: CDF of Ratio of Download Rate to Bit Rate for Video Transfers (Campus)

poor performance (i.e., slow download rate); or *poor content quality* (e.g., the viewer does not find the content interesting). Figure 2 demonstrates this quantitatively. For example, approximately 10% of the interrupted transactions had a slower download speed than encoded bit rate (as shown by ratios less than one). For these transfers, the users likely became impatient with the jerky video playback and aborted the transfer. Another 80% of interrupted transfers had ratios similar to the bulk of the completed transfers. For these we hypothesize that the users simply found the content uninteresting, and aborted the transfer some time before the end of the video.

4.2 Data Collection of Global YouTube Usage

Using a Web crawler to collect information on all of the videos present on YouTube is not a feasible (nor permitted) method for examining global YouTube file characteristics. As mentioned earlier, YouTube's video repository is considered to be the largest on the Internet, and is still growing at an estimated 65,000 videos each day [29]. Although techniques such as pacing would lessen the load of crawling on the YouTube system, crawling the entire collection of videos would still take an impractical amount of time. While methods for random sampling exist, obtaining a sufficiently large sample of videos would require placing significant load on the YouTube servers and violating the Terms of Use of YouTube.[7]

Instead, we focus on the top 100 most viewed videos of the day, week, month, and all time (as reported on YouTube) to draw insights into the relationship between videos that are globally popular and videos that are locally popular. Our choice was also motivated, in part, by empirical evidence of the Pareto principle (or the so called "80-20" or "90-10" rule) in the file referencing behaviour at Web and media servers which states 20% (or 10%) of the files on a Web server or a streaming media server accounted for 80% (or 90%) of the requests.

We utilize a two step approach to collect data on the top 100 videos on YouTube. First, each day we retrieve the pages listing the most viewed videos of the day, week, month and all time. These pages provide the video identifiers of the top 100 videos. The video identifier is an 11 character unique identifier for the video within the YouTube system. Because the top 100 video lists are spread over five pages with 20 videos on each page, each time we gather the identifiers we perform 20 page loads (five for each of the four time frames).

The second step of data collection involves using APIs that are provided by YouTube for developers.[8] The API takes the form of an HTTP GET request to a URL with a specific format. Using this format, arguments are passed indicating which API function is being called along with arguments for the function. Specifically, the "youtube.videos.get_ details" method provided in the API is used. Given a video identifier and a developer identifier (associated with a user account we created) the function returns a variety of statistics on the specified video (e.g., duration, category, ratings). This method is called for each of the identifiers collected in the first step. This results in a total of 400 API calls each time this querying is done. Since these API requests are made to the YouTube site from campus, they are included in our locally measured data. However, the probing of the most popular videos is performed at a non-peak time and contributes less than 1% to the data transferred in our study. Thus, we do not filter these requests from our dataset.

[7]YouTube Terms of Use: http://youtube.com/t/terms

Table 3: Summary of Local YouTube Data

Item	Information
Start Date	Jan. 14, 2007
End Date	Apr. 8, 2007
Total Valid Transactions	23,250,438
Total Bytes	6.54 TB
Total Video Requests	625,593
Total Video Bytes	6.45 TB
Unique Video Requests	323,677
Unique Video Bytes	3.26 TB

5. ANALYSIS

We now present a high-level analysis of data collected for this study. Summary statistics of the data collected from our edge network are presented in Section 5.1. Section 5.2 describes longitudinal characteristics observed in the YouTube traffic on our edge network. Section 5.3 discusses summary statistics from the global YouTube data.

5.1 Local YouTube Summary Statistics

We monitored YouTube traffic to and from the University of Calgary campus network for 85 consecutive days, starting on January 14, 2007 and ending on April 8, 2007. Table 3 presents summary statistics for this traffic. Our monitoring period subsumes important transitions points in the academic calendar including the beginning of the semester, the mid-semester reading break, and the last weeks of the semester; furthermore, we believe that our monitoring period is long enough to capture longitudinal changes in the characteristics of YouTube traffic.

In total we recorded 23,250,438 valid (i.e., non-failed) HTTP transactions (i.e., request/response pairs). These transactions account for approximately 6.54 TB of data transfer. Only 3% of the HTTP requests were for video files; however, the corresponding HTTP responses accounted for 99% of the total bytes transferred. Similar skewness has been observed in other types of Internet traffic; for example, Paxson observed 2% of `ftpdata` connections accounting for up to 80% of bytes transferred [36]. We also observed that over 50% of the video requests (and corresponding bytes transferred) were for previously requested videos. This indicates that in-network caching has the potential to reduce bandwidth demands for YouTube content.

Table 4 presents a breakdown of the HTTP request meth-

[8]http://www.youtube.com/dev

Table 4: Breakdown of HTTP Request Methods

Method	Total	% of Total
GET	23,221,168	99.87
POST	28,655	0.12
Others	615	0.01

Table 5: Breakdown of HTTP Response Codes

Code	% of Responses	% of Bytes
200 (OK)	75.80	89.78
206 (Partial Content)	1.29	10.22
302 (Found)	0.05	0.00
303 (See Other)	5.33	0.00
304 (Not Modified)	17.34	0.00
4xx (Client Error)	0.19	0.00
5xx (Server Error)	0.01	0.00

ods seen in the YouTube campus trace. This analysis provides insights into the activity of YouTube users on our campus network. As expected, we find that HTTP GET requests constitute the majority of requests. This indicates almost all requests are for fetching content from YouTube. We also observed 28,655 HTTP POST requests. The HTTP POST method is used by a client's browser to place content on a server. In YouTube's case, POST requests are needed to rate videos, comment on videos, and upload videos.

At first glance, the number of POSTs appears to be insignificant; however, when considered relative to the total number of video requests (625,593), POSTs are non-negligible. Note that the total number of video requests reflects how many videos were watched, and one expects user interactivity to be proportional to the frequency of use of the YouTube site. We analyzed the content-type field of the HTTP POST messages to understand the type of content that is being uploaded to YouTube.

The majority of the POSTs appear to be the result of users posting comments or rating videos. We observed only a small number of video upload attempts (133) over the three month collection period. Since our measurements are made at a campus edge network it is likely that we observe fewer uploads than would be present in other edge networks such as those that service residential users.

We believe the upload/download behaviors observed on our campus network are similar to those of other edge networks as well. For example, estimates put the number of video uploads to YouTube at 65,000 per day, compared to 100 million daily video downloads [29]. Clearly, most of the users are consumers of content and only a handful of the users are content producers, just as on our campus.

The HTTP response codes provide additional insights into YouTube's workload. The breakdown of response codes is shown in Table 5. Response code 200 indicates that a valid file was delivered to the client. Response code 206 indicates partial transfer of a file because of GET request for a specific (byte) range. Response code 304 indicates the availability of an up-to-date cached copy of the requested file in the client's cache, and is obtained in response to an If-Modified-Since request. On further analysis of the HTTP 304 responses, we find that 40% of these were generated in response to requests for JPEG files. This is not surprising as frequent visitors to YouTube are likely to retrieve many of the thumbnails from their browser's local cache. We also find that approximately 1% of the HTTP 304's were for Flash Video, which suggests some users were re-watching selected videos. HTTP response codes 200, 206, and 304 makeup 94% of the responses seen in our campus YouTube traffic. We also find

Table 6: Breakdown by Content Type (Status 200)

Item	Images	Text	Applications	Videos
Responses	13,217,449	2,020,436	1,828,486	556,353
Bytes (GB)	37.58	18.59	28.93	5,785.05
% Requests	75.00	11.46	10.38	3.16
% Bytes	0.64	0.32	0.49	98.55
File Size				
Mean (KB)	3.18	18.62	5.84	10,110.72
Median (KB)	3.17	25.76	0.22	8,215.00
COV	0.29	2.31	0.66	0.97
Transfer Size				
Mean (KB)	3.08	9.60	15.97	10,332.44
Median (KB)	3.24	7.26	21.99	8,364.00
COV	0.51	1.26	0.65	0.99

approximately 5% of the requests to be redirected to another URL (response codes 302 and 303). The 303 response codes in particular appear to be used for load balancing purposes. For example, we observed such codes in response to requests for video files on `www.youtube.com`. Each of these requests is then redirected to a different server (e.g., `v104.youtube.com`). Overall, a majority of the requests resulted in the successful delivery of the requested file to the client. Client errors (response code 4xx) and server errors (response code 5xx) are infrequently seen.

We also want to understand what types of files are transmitted as a result of campus YouTube usage. For this analysis, we categorized all HTTP 200 response messages (i.e., those responses that carried full sized content data) using information from the content-type field of HTTP responses. The results are summarized in Table 6. The results show that images (e.g., image/jpeg, image/png, image/gif) and text (e.g., text/html, text/css, text/xml) makeup 86% of all responses. Applications (e.g., application/javascript, application/xml, application/x-shockwave-flash) and videos (video/flv) account for 10% and 3% of the responses, respectively. As noted earlier, videos account for almost all (98.6%) of the bytes transferred.

The middle rows of Table 6 consider characteristics of the distinct files that were downloaded from YouTube. As one might expect, the video files are orders of magnitude larger than other file types. We also find that the mean and median sizes within each category are similar to each other. In addition, the coefficient of variation (COV) of file sizes within the image, application and video categories are less than one, suggesting the file sizes within these categories are not highly variable.

The bottom rows of Table 6 show the transfer size statistics. For Images and Videos, the transfer size statistics are quite similar to the File Size statistics. For Text, the transfers are mostly for a few smaller files, while for Applications, the transfers are mostly of a few larger files. Additional information is available in Section 6.1 and Section 9.1.

5.2 Local YouTube Utilization Characteristics

Figures 3, 4, and 5 show the number of unique YouTube users each day, the number of requests to YouTube these users generated each day, and the amount of data transferred by YouTube each day to our network, respectively.

The results show that the number of unique YouTube users increases steadily for the first three weeks thereafter increases slowly, reaching 3,000 distinct users/day in the final week of our measurement period. Correspondingly, we also observe an increase in the number of YouTube requests and the amount of YouTube bytes. There are two proba-

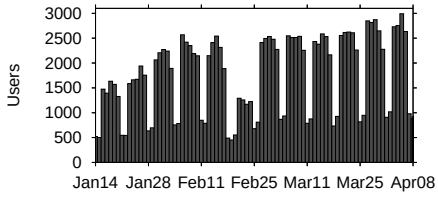

Figure 3: Unique Users Per Day

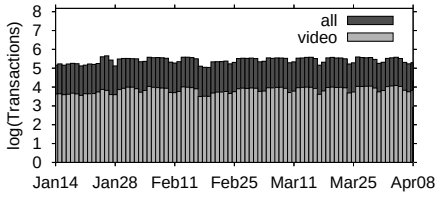

Figure 4: Requests Per Day

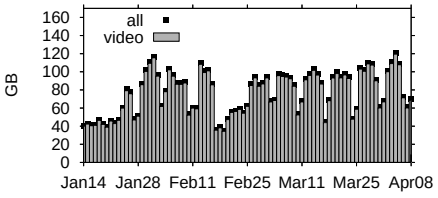

Figure 5: Bytes Per Day

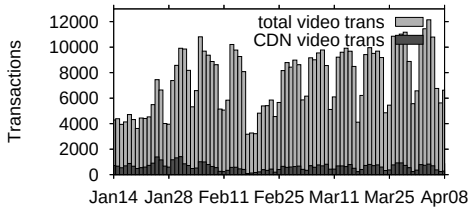

Figure 6: Video Requests Served by YouTube/CDN

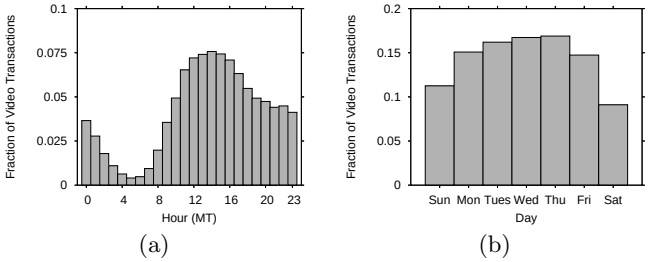

Figure 7: YouTube Traffic Patterns: (a) by time of day; (b) by day of week

ble reasons for this noticeable increase in YouTube activity in early February. First, we believe that students are more settled by early February, following the initial assignments of the semester. Second, during this time frame there was increased media coverage of YouTube. At that time, several large media companies began demanding removal of copyrighted content from the site [34]. Simultaneously, a high profile viral marketing campaign on YouTube raised awareness of the site [11]. Traffic decreases in mid-February as a result of reading break, when many students leave campus.

Figure 4 shows that the number of requests for video is approximately two orders of magnitude less than the total number of requests owing to YouTube use; however, as shown in Figure 5 video requests consistently account for almost all of the YouTube byte transfers. Because video requests account for most of the byte transfers, we focus on these requests in the remainder of this section.

Figure 6 shows how requests for videos were handled by YouTube's infrastructure. Specifically, we show how many video requests were handled by YouTube and the Limelight CDN. The graph of bytes transferred by YouTube and Limelight looks very similar to Figure 6, and is therefore omitted. We find that during our measurement period the number of requests and bytes served from the CDN on a daily basis remained fairly steady and typically accounted for less than 1,000 requests and 10 GB, respectively. It is likely that the amount of YouTube traffic transferred through the CDN network is intentionally limited, due to the cost incurred when traffic is directed to it.

Figure 7(a) shows the fraction of total video requests seen at a particular time of day, while Figure 7(b) shows the fraction of total video requests by day of week. As expected, video requests occur with higher frequencies during the weekdays than during the weekend. The time of day effects, however, are somewhat intriguing. We do observe the famous diurnal traffic pattern with more requests during day time than during night time; specifically, we find that there is a steady rise in YouTube traffic from 8 am to 1 pm, followed by a steady state of peak traffic between 2 pm and 6 pm, and subsequently, a steady decline in traffic from 7 pm to 7 am. Nevertheless, we find there is a non-negligible amount of video traffic late at night, specifically between midnight and 4 am. YouTube traffic this late at night is likely to originate from the university dormitories.

Table 7: Summary of Global YouTube Data

Time Frame	Daily	Weekly	Monthly	All Time
Unique IDs	7,515	2,288	586	149
View Count				
Average	21,085.83	139,628.08	736,081.33	5,568,708.36
Median	13,117	92,361	521,774	4,161,956
COV	1.71	1.06	0.98	0.80
Rating				
Average	4.20	3.93	3.85	4.37
Median	4.59	4.28	4.17	4.57
COV	0.24	0.23	0.24	0.16
Duration (s)				
Average	262.00	206.10	162.03	192.62
Median	182	133	138	199
COV	1.05	1.29	0.77	0.58

5.3 Global YouTube Characteristics

Table 7 summarizes statistics observed by monitoring the YouTube site, each day for 85 days, for the 100 most popular videos in the day, week, month, and all time categories. For each category, we collected 8,500 video IDs. We find that the daily top 100 list of videos changes quite often, whereas the list of videos in the monthly and all time categories change rather slowly. Our results indicate that entry into the all time category requires, on average, 8 times more views than those in the monthly category. We also find that popular videos in any of the categories considered have a high rating (e.g., 4 or more out of 5); the mean and median ratings are very similar, and the COV of the ratings is fairly low. Finally, our results indicate that the videos with longer term popularity tended to have durations well below the maximum of ten minutes. This can be seen in the mean and median values for the video durations in the weekly, monthly, and all time categories, which are in the 2.5 to 3.5 minute range. It is important to point out that the converse (that short duration videos are more likely to be popular) is likely not true, although we have not explored this.

6. VIDEO FILE CHARACTERISTICS

In this section, we characterize the YouTube video files seen in the local and global data sets. Specifically, the following characteristics are studied: file sizes, video durations, video bit rates, age of videos, video ratings, and video categories. Where appropriate, we comment on characteristics of non-video files and point out similarities as well as differ-

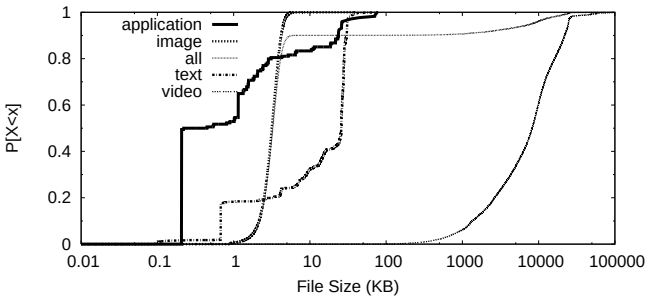

Figure 8: CDF of Unique File Sizes (Campus)

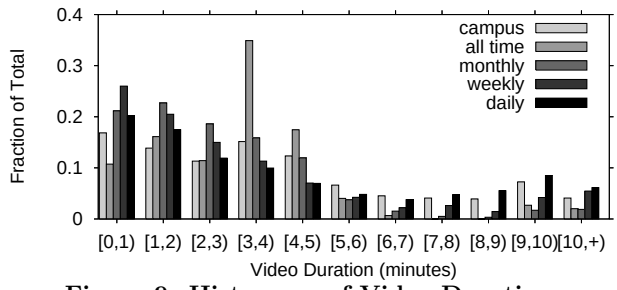

Figure 9: Histogram of Video Durations

ences with respect to traditional Web and media streaming workloads.

6.1 File Size

Unique file sizes for video and non-video content types are considered in Figure 8. Since file size is estimated using the content length field of the HTTP header, we consider only transactions with status code 200. We find that the number of unique files for image and video content types is significantly larger than the number of unique files for text and application content. We observe 2,897,298 unique files for images and 322,382 unique files for videos. In contrast, we only observe 975 unique text files and 174 unique application files. This suggests that the same framework of HTML and Javascript pages are being used to display a wide variety of images (mostly thumbnails) and videos.

YouTube's stated policy (as of this writing) is to impose a limit of 100 MB on the size of video files.[9] Nonetheless, we found a small fraction, approximately 0.1%, of the videos to be larger than 100MB, thus indicating that the file size limit is not strictly enforced. Furthermore, not many extremely large sized video files appear to be posted and/or accessed by campus users; only 10% of the videos requested are larger than 21.9 MB. We find that unique file sizes for video are orders of magnitude larger than those observed for other content types. These larger files will require more storage space than traditional text based Web content.

6.2 Video Duration

In this section we analyze the duration of video files seen in our traces. Durations for the globally popular videos were retrieved using the YouTube API, as described in Section 4.2. Since our local data collection process does not provide the duration of each video, we also used YouTube's API to obtain this information for data collected locally. Figure 9 shows a histogram plot of the video durations in each of the different categories.

[9]http://www.google.com/support/youtube/bin/answer.py?answer=55743& topic=10527

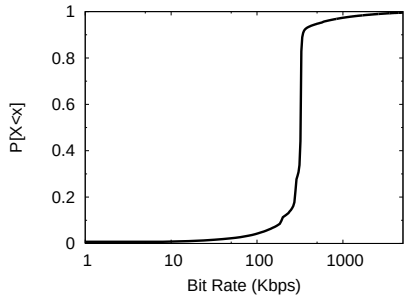

Figure 10: CDF of Video Bit Rates (Campus)

YouTube places a cap of 10 minutes on video length.[9] However, users with "director" accounts are able to post content that is longer than 10 minutes. In our analysis, we noticed a few videos which significantly exceeded the 10 minute limit. Specifically, we observed a video that reported a length of over 60,000,000 seconds. Clearly, this is not a valid video length. The user with this misreported video length had other misreported durations in their uploaded videos. We are unable to determine the precise cause of these incorrect video durations but suspect it occurs when the video is converted from its original format into Flash Video. In order to limit the impact of these incorrect video lengths, we focus our analysis on videos with lengths of less than 2 hours. This captures 99.9% of the videos observed on campus during our measurement period. Not including videos that are longer than 2 hours, we find that the mean video duration observed on campus is 4.15 minutes with a median of 3.33 minutes. The COV is approximately 1.

Figure 9 also shows that videos with longer-term popularity tend to be shorter than others. For example, we find that 52.3% of the videos in the all time popular category are between 3 and 5 minutes long. Compared to videos in the all time category, we find longer duration videos in the daily and weekly popular categories. Table 7 shows that as the time frame of popularity increases we observe a decrease in the coefficient of variation from 1.29 in the weekly most popular list to 0.58 in the all time most viewed list. This decrease in variability is also evident in the spike in the histogram for all time most popular videos between 3 and 4 minutes.

Our analysis indicates that YouTube videos are slightly longer than videos found on the Web by Li *et al.* [28]. Their study had found that the median size of video clips on the Web was about 2 minutes.

6.3 Bit Rate (Campus)

The encoded bit rate of a video is an indicator of its playback quality. Understanding if the bit rate (and thus playback quality) is too low is of interest for several reasons. First, the popularity of YouTube might decline over time if other video sharing sites offered videos encoded at a higher bit rate. Second, video file sizes might increase in the future, if higher bit rates are demanded by users.

Unfortunately, the bit rate information is not readily available for YouTube videos. However, for the videos accessed on our campus network, we were able to estimate the encoded bit rate as the ratio of a video's file size (obtained from the `Content-Length:` header) and its duration (retrieved using the YouTube API). The results are shown in Figure 10.

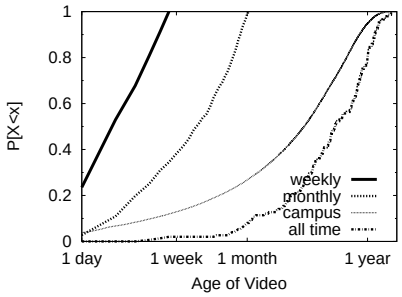

Figure 11: CDF of Age of Video Content

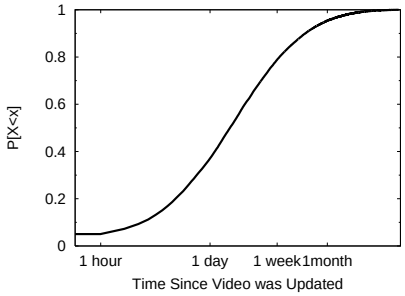

Figure 12: CDF of Time Since Video Update (Campus)

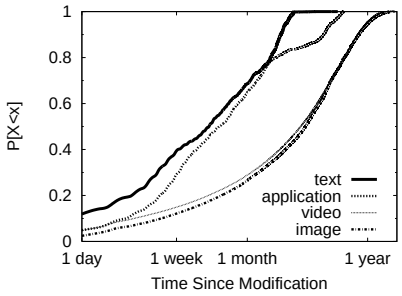

Figure 13: CDF of Time Since File Modification (Campus)

From Figure 10 several observations can be made. We find that, among the videos accessed, only a very small number are encoded at extremely low bit rates (e.g., 10's of Kbps). This suggests dial-up users are not the target audience. Similarly, we find very few videos encoded at high bit rates (e.g., above 1 Mbps). The mean and median bit rates of the videos accessed on campus was 394 Kbps and 328 Kbps, respectively. Approximately 97% of the videos seen on campus have bit rates below 1 Mbps. A large number of the videos, 62.6%, have bit rates between 300 Kbps and 400 Kbps. Our results show that most videos are encoded to enable the typical broadband user to begin playback with minimal startup delay. It is interesting to compare our results with those of Li *et al.* [28] who had found the median bit rate of stored video files on the Internet to be around 200 Kbps, with approximately 30% of the content encoded at less than 56 Kbps. Our results show that YouTube bit rates are somewhat higher than those reported for on-demand streaming in earlier work, possibly due to the improved broadband connectivity of the end users.

6.4 Age of Videos

Since YouTube (following the Web 2.0 model) allows all users to publish videos to their site, there is always new content to be viewed. In this section, we investigate how old content consumed by users is. The first measure we consider is the age of videos. We define the *age* of a video as the difference between the time the video was uploaded (gathered from the API) and when the video was retrieved from YouTube (or observed on a most viewed list in the case of globally popular videos).

Figure 11 graphs the age of videos in the weekly, monthly, and all time most viewed lists as well as the age of videos viewed on campus. Note that videos in the daily most popular list tend to be less than 3 days old and are not shown on the graph. As expected, we observe that videos in the weekly and monthly most viewed lists tend to be under 1 week or 1 month old, respectively. In contrast, videos in the all time most viewed videos tend to be older. Interestingly, we also observe older videos on campus where 73% of videos are over 1 month old and 5% are over 1 year old. This suggests that users on campus enjoy content that has been around for a while.

To further investigate how "current" videos viewed by campus users are, we consider how long it has been since a viewed video has been "updated". An update may include user interactions with a video such as adding comments, etc. The time a video has been updated can easily be retrieved using the YouTube API. Figure 12 shows the empirical distribution of the time since a video has been updated (in relation to when it was retrieved) for videos viewed on campus. While videos viewed on campus are generally not the most recent content, they are usually recently updated. We find that 95% of videos viewed on campus have been updated in the last month.

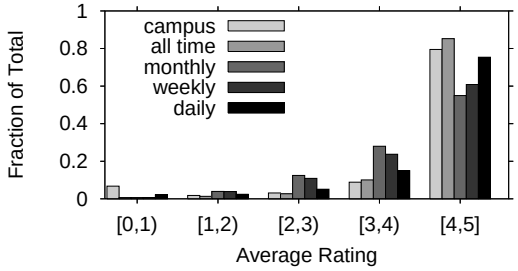

Figure 14: Histogram of Average Video Ratings

The time since modification of a file is defined as the difference between the time a file was last modified (retrieved from the HTTP header) and the time it was served to the user. The time since modification is an important measure of content age to study as it directly impacts the effectiveness of caching where out of date content requires refetching. In Figure 13 we consider the time since modification for various content types on campus. We observe that video and image files remain unmodified for longer periods, with 50% of videos not being modified in the past 89.9 days and 50% of images not being modified in the past 99.3 days. Application and text files are updated more frequently, with 50% of text files being modified in the past 13.7 days and 50% of application files being modified in the past 16.8 days. This implies that relative to application and text content, videos and images remain fairly static, thus requiring less refetching to keep them up to date.

6.5 Rating of Videos

An important part of Web 2.0 is user interaction. One of the interactive features of YouTube is a video rating system where users may rate videos on a scale of 0-5 "stars" (0 being low and 5 being high). The average rating of a video provides insight into how well liked it is by users. In this portion of our characterization of YouTube traffic we examine whether users enjoyed the content they were watching. The answer to this question is generally yes, as illustrated in Figure 14 where we present a histogram of ratings for unique videos.

For all sets of videos we observed, the average rating is 3 or higher over 80% of the time. The mean rating of videos in the most popular lists is consistently near 4 with very little variation. We make similar observations on campus where

Table 8: Summary of Video Categories

Category	Campus	All Time	Month	Week	Day
Autos & Vehicles	2.56	0.79	3.01	2.67	1.94
Comedy	**13.60**	**25.40**	**18.88**	**13.90**	**10.36**
Entertainment	**23.97**	**22.22**	**21.69**	**19.31**	**20.46**
Film & Animation	7.05	7.14	5.62	5.23	6.70
Gadgets & Games	4.09	0.79	2.81	4.93	6.72
Howto & DIY	2.38	0.00	1.61	2.91	2.02
Music	**22.35**	**30.95**	**20.28**	**11.88**	**9.57**
News & Politics	*3.34*	*3.17*	*5.42*	*9.92*	*10.02*
People & Blogs	6.09	5.56	10.04	9.98	8.72
Pets & Animals	1.87	3.17	1.81	1.84	1.19
Sports	*11.26*	*0.00*	*7.43*	*16.64*	*21.69*
Travel & Places	1.45	0.79	1.41	0.77	0.62

the mean rating is 4.18 and the coefficient of variation is 0.32.

As YouTube is an ever expanding and enormous video library, it is certainly very difficult to browse through all available content and find which ones to watch. Therefore, one might expect ratings to aid users find content of interest among the large volume of content available at YouTube.

6.6 Video Category

The myriad videos available from YouTube are categorized by YouTube into 12 categories, ranging from Autos & Vehicles to Travel & Places. All 12 categories are listed in Table 8. We note that all of the categories we consider existed for several months before our measurement period. In this section we investigate the types of videos people are watching on YouTube. We do this utilizing information from YouTube's API. Table 8 summarizes the percentage of videos observed in each category, both on campus as well as in the most popular (global) lists.

We find that in the daily and weekly data sets, popularity of categories is more uniform than in longer time frames where clear peaks emerge, specifically around comedy, entertainment, and music (shown in bold). What is popular in the different time frames also varies. On a daily basis, entertainment and sports are most popular, followed by news and comedy. This suggests daily popular events may center around current events in news and sports (shown in italics). As the time frame considered increases, we observe most of the videos are comedy, entertainment, and music. Because these types of content are often enjoyable regardless of their recency they lend themselves well to being viewed a large number of times. On campus we observe similar trends, with the top 4 categories being, entertainment, music, comedy, and sports.

It is also interesting to note which categories are not popular. In most cases, the least popular categories are Autos & Vehicles, Howto & DIY, Pets & Animals and Travel & Places. The nature of these categories suggests users viewing videos on the YouTube Web site are looking for entertainment rather than reference information on specific topics. This is in contrast to other Web 2.0 Web sites such as Wikipedia where users are usually looking for information.

7. FILE POPULARITY (CAMPUS)

File popularity has important implications for systems design and planning. In this section we consider two different approaches to analyzing file popularity, namely Zipf analysis and concentration analysis, to understand the video referencing behaviour of YouTube users on our campus.

7.1 Zipf Analysis

Zipf's law states that if objects are ranked according to the frequency of occurrence, with the most popular object assigned rank of one, the second most popular object assigned a rank of two, and so on, then the frequency of occurrence (F) is related to the rank of the object (R) according to the relation,

$$F \sim R^{-\beta}$$

where the constant β is close to one [44]. Zipf's law has previously been used to model Web document references [7, 8, 31] and media file references [14, 15, 38, 43].

The simplest verification of the applicability of Zipf's law is to plot the rank ordered list of objects versus the respective frequency of the object on a log-log scale. On a log-log scale, the observance of a straight line is indicative of the applicability of Zipf's law. The plot in Figure 15 shows that video references at our campus follow a Zipf-like distribution. We determined the exponent β by performing a regression analysis. We find $\beta = 0.56$ fits our empirical observations very well with an R^2 goodness of fit value of 0.97. This β value is slightly lower than the values reported by Breslau *et al.* [8] and Mahanti *et al.* [31] for Web proxy workloads (0.64-0.83).

Two factors contribute to the observed Zipf-like behavior. First, we believe that some of the YouTube content viewed on campus is genuinely popular among multiple users. Another potential factor is YouTube's infrastructure which aims to disallow downloading of videos. As a result, users wishing to view the same content again must return to YouTube and issue another request.

7.2 Concentration Analysis

Another approach to understanding how skewed the references are toward certain videos is the concentration analysis. The objective of this analysis is to determine the fraction of the total references accounted for by the most popular videos. This technique of analyzing skewness in the referencing behaviour was applied previously to understand memory and file referencing behaviour [10, 32], Web document referencing behaviour [7, 31], and more recently to the referencing behaviour of media files on an on-demand streaming system [43].

Figure 16 shows the cumulative distribution of the number of references and corresponding bytes for videos which are sorted in descending order according to their observed frequency of reference. We find that for video requests made by the campus community this principle does not hold. In fact, the top 10% of videos only account for 39.7% of the videos and the top 20% account for 52.4%. Clearly, the Pareto rule (discussed earlier) which was observed in Web and media server workload studies [7, 31, 43], is generally not applicable for the campus YouTube video workload. However, our observed video request pattern is similar to file access patterns of Web proxy workloads, as one would expect given the lower β values [8, 31].

We also analyzed occurrence of one-timer videos, that is videos that are requested only once in the entire data collection period. We found that 220,389 one-timer videos. These one-timers account for 68.1% of the videos and 35.3% of the total video requests, respectively. In terms of bytes, one-timers account for approximately 13.6% of the total video bytes transferred. In a similar analysis of Web doc-

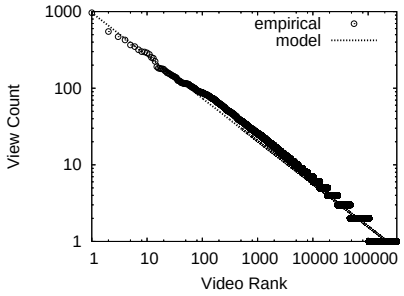

Figure 15: Ranked View Count of Videos (Campus)

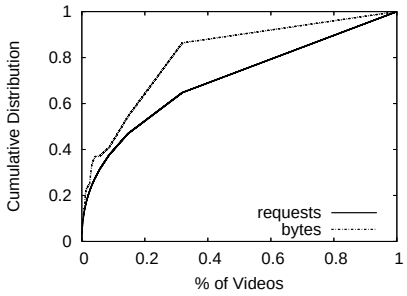

Figure 16: Concentration of Video References (Campus)

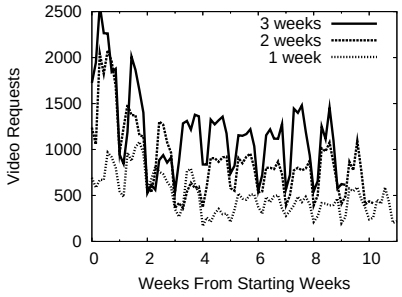

Figure 17: Absolute Drift From First Weeks

uments, it was found that approximately $15-30\%$ of the documents referenced at a Web server and approximately $70-75\%$ of the documents referenced at a Web proxy are one-timers [6, 7, 31].

A plausible explanation for why we do not observe the Pareto rule in our video workload is the diversity of content available on YouTube. YouTube offers many more (probably several orders of magnitude more) videos than traditional media-on-demand servers analyzed in the literature. More choices may translate into fewer requests per video as videos become more specialised and have more limited audiences (e.g. home videos). The effects of the large amount of available content are amplified by our edge network point of view. At the central YouTube server the amount of content is quite large, but so is the user population. At an edge network, the number of users is low when compared with the number of global users. This smaller population still has access to the large repository of content available on the YouTube site, likely resulting in less concentration in file accessing behavior. Similar observations have been made for Web proxy workloads [31].

8. LOCALITY CHARACTERISTICS

In this section, we consider the temporal locality characteristics of YouTube videos accesses on the campus network. Temporal locality is the idea that events in the recent past are good indicators of events in the near future. This principle has been applied in operating systems where it has been found that memory blocks referenced by a program in the immediate past and near future exhibit high correlations [19]. Similarly, locality has also been found to occur in Web server and proxy document reference streams [7,30,31]. In this section, we consider temporal locality using working set analysis, as has been applied in a Web context. We also examine locality between the most popular videos on YouTube and videos that are viewed on campus.

8.1 Working Set Analysis

Working set analysis is often used to understand how popularity of objects changes with time. We consider absolute drift in the working set relative to the first weeks in Figure 17. We observe that the number of requests in common with the first weeks is sensitive to the lower request frequencies that we observe on weekends. However, during the week when there are more requests we observe more similarity between the first weeks and the daily requests. When considering the set of videos observed in the first week, we find that approximately 500 of the videos persist throughout our measurement period. For the sets of videos observed in the first 2 and 3 weeks we observe approximately 900 and 1200 persistent videos, respectively.

Figure 18 considers short term temporal locality in the set of videos viewed each day (working set). We find that there is not a very strong correlation between videos viewed on consecutive days. In general, 10% of the previous days videos are viewed again on the following day. An interesting trend in our working set analysis is similarity between the number of videos viewed on a given day and the observed short term temporal locality. At the beginning of our measurement period when there is less traffic, temporal locality is usually close to 5%. However, as interest in YouTube increased in early February we noticed a rise in temporal locality to 10%. A similar trend is evident on weekends when video accesses are less numerous. It is possible that if YouTube traffic were to increase again, commonality between consecutive days may also increase, making day to day caching a viable strategy for limiting the impact of YouTube on network resources.

Absolute growth in the working set is considered in Figure 19. We observe that the number of videos viewed on campus increases faster than the set of unique videos that are observed. By the end of our trace period the total number of videos viewed is 625,593 whereas the number of unique videos viewed is 323,677. This large difference between unique content and total content suggests that if a cache were allowed to cache all video content for an indefinite period of time, the savings in network bandwidth resources could be significant (a factor of 2 reduction in our case). As is seen in Table 3, this would translate into a savings of 3.19 TB.

8.2 Global Versus Local Popularity

From a service provider's perspective, global activity is often of greater importance than local activity. However, as was the case with this study, the availability of information about global activity may be limited or even non-existent. In this section, we examine what global information we might infer by studying edge network activity.

We analyze the relationship between global popularity and files that are viewed on campus in Figure 20. We find that approximately half of the top 100 videos are viewed on campus; however, they do not contribute significantly to the total videos viewed on campus on a daily basis. On most days the popular videos account for less than 1% of the videos viewed on campus. This may be as a result of users not browsing these most viewed lists when they are visit-

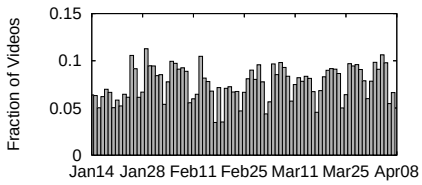

Figure 18: Fraction of Previous Days Video Requests Observed

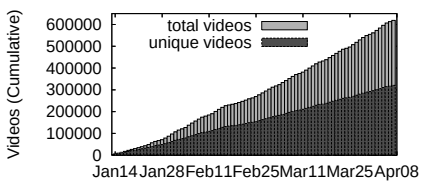

Figure 19: Unique and Total Video Growth

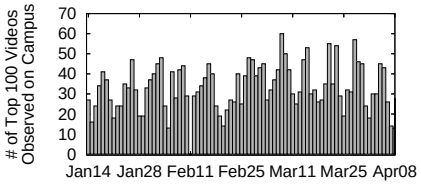

Figure 20: Overlap Between Globally Popular and Campus Videos

ing YouTube. It may be the case that users are directed to YouTube by friends sending them specific videos, rather than going there to browse the large repository of videos. With the recent surge in popularity of Web 2.0 social networking sites such as Facebook which allow users to embed YouTube videos into their profile page, it is likely that the number of users who browse the most popular lists will remain low while the number of users in general will increase.

9. VIDEO TRANSFER CHARACTERISTICS

9.1 Transfer Size

We analyzed the transfer sizes of video and non-video content accessed from YouTube by our campus clients. Transfer sizes are also estimated using HTTP responses content length field. Estimation is required because of gapped transmissions, where calculating the amount of data transferred using TCP sequence numbers is not possible. Consequently, we restrict attention to HTTP responses containing full size content (i.e., status code 200).

Figure 21 presents the cumulative distribution of video and non-video transfer sizes. Similar to file sizes, we observe video content transfers that are orders of magnitude larger than transfers for non-video content from the YouTube site. Video transfer sizes range from very small to very large values. Typically, the small sized transfers represent short duration video clips and the large size transfers represent long duration video clips.

Most of the images transferred from YouTube are JPEG thumbnails that appear on almost every page of the YouTube site. Our results suggest that these images are typically less than 5KB in size. Surprisingly, the text transfers (e.g., HTML, CSS, and XML files) are larger than the images. Many Web 2.0 sites, including YouTube, are using Asynchronous Javascripts and XML (AJAX) techniques to design interactive Web sites. Typical use of AJAX involves bundling Javascript with HTML, which is likely the reason why we observe transfers of text files that are generally larger than images.

A spike is observed in transfer size distribution for application content around 26 KB. We have verified that this spike is caused by transfer of a SWF media player file (e.g., `player2.swf`, `p.swf`). Steps in the lower portion of the graph are due to transfers of Javascript objects. These Javascripts are used for tasks such as managing comments, the rating system, and embedding the flash player.

9.2 Transfer Duration

In the preceding section, we observed that video content transfer sizes are orders of magnitude larger than non-video content served by YouTube. These larger transfers not only require increased storage capacity at servers, but also more processing power to handle the longer durations required to transmit the larger content. Figure 22 shows the CCDF of transfer durations for the various types of content served by the YouTube site. We observe that video transfers have durations that are orders of magnitude larger than other content types. While text, image and applications have median durations of less than 1 second, video transfers exceed 1 second 96.6% of the time. The mean transfer duration for video content is 104.4 seconds, which is orders of magnitude larger than the means observed for the other content types. The longer time required for transferring video content implies that as YouTube becomes more popular, more processing power will be required at servers to handle multiple concurrent requests for video content.

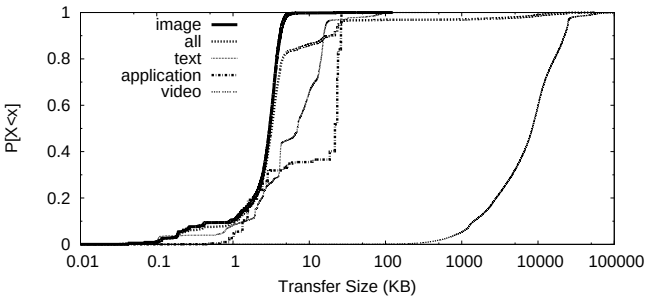

Figure 21: CDF of Transfer Sizes (Campus)

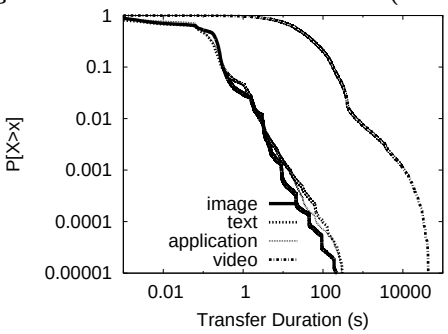

Figure 22: CCDF of Transfer Duration (Campus)

10. DISCUSSION

In this section we describe the significance of the results in Sections 5 through 9. In Section 10.1 we discuss issues for (edge) network providers. In Section 10.2 we examine implications for service providers.

10.1 Implications for Network Providers

The most obvious issue created for network providers by Web 2.0 is the increased bandwidth consumption for transporting large multimedia objects (e.g., videos, high resolution photos). Caching and CDNs, two solutions utilized for "traditional" Web workloads, are also suited to Web 2.0 workloads, although some differences exist. We examine each of these potential solutions individually.

Web caching emerged in the mid 1990's as an approach for reducing the bandwidth consumption of network links, reducing the load on origin servers, and improving the end user experience by reducing the retrieval times of static Web objects. Over time numerous incremental improvements were made, such as enhanced (cached) object replacement algorithms and cache consistency techniques. Since Web 2.0 utilizes the same application layer protocol (HTTP) as the "traditional" Web, existing Web caching infrastructures can benefit Web 2.0 as well. However, such infrastructures may not be optimally suited for Web 2.0 workloads, and utilizing the same infrastructure for both may degrade the performance for both. We are not suggesting that a separate physical infrastructure is needed, but separate logical infrastructures may provide both with performance isolation from each other.

There are a number of reasons Web 2.0 workloads could be treated differently from traditional Web workloads. First, the number of large multimedia files is likely to be much greater for Web 2.0 workloads. These files will account for the majority of the bytes transferred over the network, even if they are only a small percentage of the total requests. In order to reduce (peak) bandwidth consumption, more of these objects must be cached, which may displace many smaller objects. This could degrade the experience for many other users, as cache hit rates decrease. Second, larger cache sizes may be required, as the breadth in interests may require many more objects to be cached in order to achieve a reasonable hit/byte hit rate. Third, the object replacement algorithm of choice may differ; while some characteristics (e.g., recency, frequency) may still be important, others (e.g., size) may be less useful. In addition, the additional meta-data available (e.g., user ratings, content topic, etc) with Web 2.0 applications will provide important information, and should be exploited to improve the effectiveness of caching algorithms.

In the late 1990's, Content Distribution Networks (CDNs) emerged. CDNs provided many of the benefits of Web caching, and also gave content providers more control over their content. In particular, CDNs enabled a provider to improve the overall browsing experience for their users. CDNs are a potential alternative to Web 2.0 caching for edge network providers. For example, our campus hosts nodes from at least one CDN; requests for files on this CDN can be served locally and generate little or no traffic on our external Internet link. If YouTube traffic became significant enough on a network link (it is currently responsible for 4.6% of traffic on our campus Internet link), a network provider could consider hosting one or more nodes for the Limelight CDN. Due to the breadth of content, and the reduced concentration of reference we have shown, prefetching/preloading techniques to populate the storage on a CDN node may be relatively ineffective. At the very least, such techniques will need to leverage the meta-data that is available, and place greater importance on local interest than global popularity.

10.2 Implications for Service Providers

A fundamental difference between the traditional Web and Web 2.0 is that the content creation process is now widely distributed, and is (mostly) independent of the content hosting (which is done by a service provider such as YouTube). This difference has several implications for the service provider, who must plan, purchase, install, operate and maintain the central infrastructure used by the site. Two important issues are storage and computation requirements; we discuss each in turn.

User interest in multimedia content is not new; what has changed is the availability of content. In the traditional Web, the availability of interesting multimedia content was often limited (free or otherwise). Only in recent years has multimedia content began to appear online, as business models were put in place (e.g., iTunes). With Web 2.0, social networking effects can result in large user communities growing around a service. Given the relative ease with which a person can now create digital content (text, photos, videos, etc.), coupled with human interest in retaining such information indefinitely, it seems that there is sustainable demand for continued growth in storage capacity. For example, YouTube receives an estimated 65,000 new videos per day [29]; with an average size of 10 MB for each video (Table 6), this means YouTube's video repository grows by approximately 19.5 TB per month! Furthermore, if the user base increases, a larger number of users start to contribute content, or if longer/larger videos are permitted, the rate of growth could increase further. In addition, since much of the content is likely to be unpopular (the *long tail effect*), it will be important to minimize the cost for storing that content. This suggests high capacity, low cost disks (e.g., SATA) with less redundancy than might be used for hosting traditional Web sites.

Workloads for sites such as YouTube also have implications for the choice of server used to operate the service. For example, serving large objects such as videos utilizes more CPU cycles and takes a longer duration to complete than serving small (static) objects. Since servicing large transactions can occupy an HTTP server process or thread for longer periods of time, this can limit the concurrency of the server. Tuning the appropriate parameters on the HTTP server is only one issue to consider. Such workloads should be better suited to multi-core systems than traditional single-core systems, which can better support large numbers of processes or threads in parallel. In addition, large memory configurations may improve performance, as the working sets are large. I/O performance will also be important, as the breadth of requests (and available content) means many requests will be served from disk.

11. CONCLUSIONS

With the rise of Web 2.0 technologies on the Web, there is a need to understand their workload patterns, in order to plan, design, and build more efficient delivery infrastructures. Popular Web 2.0 sites, such as YouTube and Flickr, support multiple authors posting and sharing large media files; this places significant demand on server and network resources.

We examine the immensely popular video sharing Web 2.0 site, YouTube. The popularity of YouTube, combined with the amount of data transferred by it, poses many challenges to measurement of its long-term behavior. To address these challenges, we take a multi-level approach to measurement, observing YouTube traffic locally in a campus setting as well as examining over time the most popular videos on the site.

After an extensive analysis of the YouTube workload, we find that there are (not surprisingly) many similarities to traditional Web and media streaming workloads. For example, access patterns are strongly correlated with human be-

haviors, as traffic volumes vary significantly by time-of-day, day-of-week, as well as longer term activities (e.g., academic calendars). Similarly, video files are much larger than files of other types, and some videos are more popular than others. These and other characteristics suggest that caching should improve the performance and scalability of Web 2.0. However, there are differences as well. In particular, enabling anyone (and everyone) to publish content means growth in content will not only be larger than for traditional Web and media, but sustainable. This will place greater strain on centralized resources, and require decentralized approaches such as caching and CDNs. Furthermore, the breadth and depth of available content reduces the concentration of references in the access stream, which can reduce the effectiveness of caching and prefetching strategies. However, the increased availability of meta-data in Web 2.0 (a direct result of social networking) can and should be exploited to make such techniques more effective.

As future work we plan to upgrade to a more powerful monitor platform. This should enable us to monitor all of the campus Internet traffic, not limit us to a static set of IP addresses, and reduce or potentially eliminate the "gapped" transactions. Our plan is to decompose all of our campus Internet traffic, with a focus on interesting new applications such as YouTube, MySpace, and Flickr. As we develop a `bro` script for this work, we intend to reduce the number of connections that fail to parse, if at all possible. If feasible from an overhead and privacy perspective, we will also collect additional information; in particular, HTTP headers such as `Cache-Control:` and `Location:`.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers and our shepherd, S. Machiraju, for their insightful comments that have helped to improve this paper. This work was supported by the Informatics Circle of Excellence (iCORE) in the Province of Alberta and the Natural Sciences and Engineering Research Council (NSERC) of Canada.

12. REFERENCES

[1] About the U of C. http://ucalgary.ca/about/.
[2] S. Acharya and B. Smith. An Experiment to Characterize Videos Stored on the Web. In *Proc. SPIE/ACM MMCN*, San Jose, USA, Jan. 1998.
[3] S. Acharya, B. Smith, and P. Parnes. Characterizing User Access to Videos on the World Wide Web. In *Proc. SPIE/ACM MMCN*, San Jose, USA, Jan. 2000.
[4] J. Almeida, J. Krueger, D. Eager, and M. Vernon. Analysis of Educational Media Server Workloads. In *Proc. ACM NOSSDAV*, Port Jefferson, USA, June 2001.
[5] M. Ames and M. Naaman. Why We Tag: Motivations for Annotation in Mobile and Online Media. In *Proc. ACM CHI*, San Jose, USA, May 2007.
[6] M. Arlitt and T. Jin. Workload Characterization of the 1998 World Cup Website. *IEEE Network*, 14(3), 2000.
[7] M. Arlitt and C. Williamson. Internet Web Servers: Workload Characterization and Performance Implications. *IEEE/ACM Trans. on Networking*, 5(5), 1997.
[8] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. IEEE INFOCOM*, New York, USA, Mar. 1999.
[9] Bro. http://www.bro-ids.org.
[10] R. Bunt and J. Murphy. The Measurement of Locality and the Behaviour of Programs. *Computer Journal*, 27(3), 1984.
[11] CBC. YouTube's Bride Wig Out Revealed as 'net seed' for Ad Campaign. *CBC Arts*, Feb. 2007.
[12] E. Chang, M. Davis, P. Schmitz, and S. Boll. Panel Discussion: Web 2.0 and Multimedia: Challenge, Hype, Synergy. In *Proc. ACM MULTIMEDIA*, Santa Barbara, USA, Oct. 2006.
[13] X. Cheng, C. Dale, and J. Liu. Understanding the Characteristics of Internet Short Video Sharing: YouTube as a Case Study. Technical Report arXiv:0707.3670v1 [cs.NI], Cornell University, arXiv e-prints, July 2007.
[14] L. Cherkasova and M. Gupta. Analysis of Enterprise Media Server Workloads: Access Patterns, Locality, Content Evolution, and Rate of Change. *IEEE/ACM Trans. on Networking*, 12(5), 2004.
[15] M. Chesire, A. Wolman, G. Voelker, and H. Levy. Measurement and Analysis of a Streaming Media Workload. In *Proc. USITS*, San Francisco, USA, Mar. 2001.
[16] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Ribeiro-Neto. Analyzing Client Interactivity in Streaming Media. In *Proc. WWW*, NY, USA, May 2004.
[17] M. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Trans. on Networking*, 5(6), 1997.
[18] C. Cunha, A. Bestavros, and M. Crovella. Characteristics of World Wide Web Client-based Traces. Technical Report BUCS-TR-1995-010, Boston University, USA, Apr. 1995.
[19] P. Denning. Working sets past and present. *IEEE Trans. Software Eng.*, 6(1), 1980.
[20] B. Duska, D. Marwood, and M. Freeley. The Measured Access Characteristics of World-Wide-Web Client Proxy Caches. In *Proc. USITS*, Monterey, USA, Mar. 1997.
[21] Facebook. http://www.facebook.com.
[22] Flickr. http://www.flickr.com.
[23] FlixHunt. http://www.flixhunt.com.
[24] S. Gribble and E. Brewer. System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace. In *Proc. USITS*, Monterey, USA, Mar. 1997.
[25] L. Guo, E. Tan, S. Chen, Z. Xiao, O. Spatscheck, and X. Zhang. Delving into Internet Streaming Media Delivery: A Quality and Resource Utilization Perspective. In *Proc. ACM IMC*, Rio de Janeriro, Brazil, Oct. 2006.
[26] M. Halvey and M. Keane. Exploring Social Dynamics in Online Media Sharing. In *Proc. of WWW*, Banff, Canada, May 2007.
[27] N. Harel, V. Vellanki, A. Chervenak, G. Abowd, and U. Ramachandran. Characterizing A Media-Enhanced Classroom Server. In *Proc. of IEEE Workshop on Workload Characterization (WCC)*, Austin, USA, Oct. 1999.
[28] M. Li, M. Claypool, R. Kinicki, and J. Nichols. Characteristics of streaming media stored on the web. *ACM Trans. Inter. Tech.*, 5(4), 2005.
[29] Business Intelligence Lowdown. Top 10 Largest Databases in the World, Feb. 2007.
[30] A. Mahanti, D. Eager, and C. Williamson. Temporal Locality and its Impact on Web Proxy Cache Performance. *Perform. Eval.*, 42(2-3), 2000.
[31] A. Mahanti, C. Williamson, and D. Eager. Traffic Analysis of a Web Proxy Caching Hierarchy. *IEEE Network*, 14(3), 2000.
[32] S. Majumdar and R. Bunt. Measurement and Analysis of Locality Phases in File Referencing Behaviour. In *Proc ACM SIGMETRICS/PERFORMANCE*, Raleigh, USA, June 1986.
[33] J. Milani. Coming to Your Screen: DIY TV. *BBC Money Programme*, 2007.
[34] M. Musgrove. Viacom Decides YouTube Is a Foe. *Washington Post*, Feb. 2007.
[35] My Space. http://www.myspace.com.
[36] V. Paxson. Empirically-Derived Analytic Models of Wide-Area TCP Connections. *IEEE/ACM Trans. on Net.*, 2(4), 1994.
[37] P. Schmitz. Leveraging Community Annotations for Image Adaptation to Small Presentation Formats. In *Proc. ACM MULTIMEDIA*, Santa Barbara, USA, Oct. 2006.
[38] K. Sripanidkulchai, B. Maggs, and H. Zhang. An Analysis of Live Streaming Workloads on the Internet. In *Proc. ACM IMC*, Taormina, Italy, Oct. 2004.
[39] Tcpdump. http://www.tcpdump.org.
[40] USA Today. YouTube Serves up 100 million Videos a Day Online, July 2006.
[41] Wordpress. http://www.wordpress.com.
[42] YouTube. http://www.youtube.com.
[43] H. Yu, D. Zheng, B. Zhao, and W. Zheng. Understanding User Behavior in Large-Scale Video-on-Demand Systems. *SIGOPS Oper. Syst. Rev.*, 40(4), 2006.
[44] G. Zipf. *Human Behavior and the Principle of Least Effort.* Addison-Wesley (Reading MA), 1949.