UNIVERSITY OF CALGARY

YouTube Workload Characterization

by

Phillipa Gill

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA
March, 2008

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "YouTube Workload Characterization" submitted by Phillipa Gill in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE.

Supervisor
Dr. Zongpeng Li
Department of Computer Science
University of Calgary

Dr. Diwakar Krishnamurthy
Department of Electrical and Computer Engineering
University of Calgary

Co-supervisor
Dr. Anirban Mahanti
Department of Computer Science and Engineering
Indian Institute of Technology

Dr. Carey Williamson
Department of Computer Science
University of Calgary

Date

Abstract

Recent developments in Web technology have changed the way people use the Web. This thesis presents one of the first workload characterizations of YouTube, an extremely popular video sharing Web site. Over a three month period, data was collected on campus YouTube usage, including over 23 million transactions. Statistics on the globally popular videos on YouTube were also collected during this time.

The characterization presented in this thesis considers properties of YouTube traffic at aggregate, user, and session levels. We find that video content is much larger and has longer transfer durations than traditional Web content. We also observe that the popularity of videos viewed on campus is not strongly concentrated on a small number of videos. At the session level, it is observed that inter-transaction times are impacted by the driving technology of Web 2.0, `AJAX`. The size of video content also impacts the amount of data sent within user sessions. The results of this thesis are of interest to network and service providers as well as individuals developing models of Web traffic.

Acknowledgments

I would like to extend gratitude to the people who have helped me during the completion of this thesis. First, I would like to thank my advisors Zongpeng Li and Anirban Mahanti. Their guidance and encouragement have made this thesis possible. Their attention to detail has helped to improve this work.

Martin Arlitt also deserves many thanks. His expertise has helped improve the quality of this thesis and taught me a great deal about Internet measurement. This project would not have been possible without the many hours he spent collecting data.

I would also like to thank my thesis committee members, Diwakar Krishnamurthy (external) and Carey Williamson. Their detailed comments and suggestions have helped to improve this thesis.

Many thanks go to the administrative staff in the computer science department. Their patience with my many questions has been greatly appreciated.

I would like to thank my husband Brendan for encouraging me during the course of my undergraduate and graduate studies. He has been understanding, even when research occupied a lot of my time.

Finally, I would like to thank my parents. They have taught me the value of hard work which has been important for completing this project.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Since its creation, the Internet has undergone many changes. From a backbone network used to connect researchers (ARPAnet), the Internet evolved into a highly commercial entity with commercial Internet Service Providers (ISPs) providing links to connect regional networks [40]. Concurrent with this commercialization of the Internet infrastructure was the development of the Internet's first "killer" application, the World Wide Web (the Web). The Web provided a platform for many popular services such as online news reports, banking, and streaming video [40]. As the Web developed in the late 1990's and early 2000's, some interaction became possible on Web sites. This interaction came primarily in the form of discussion forums or chat applications. Other than these interactive tools, however, most Web sites still provided content from a single author with a large user base viewing this content. This notion of a single author with many users viewing content led to the development of techniques to handle the demands of Web users such as caching and content distribution networks.

In recent years, there has been a shift in how people use 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 on the Web. Content published by users, referred to as *user generated content*, can take many forms. They include textual information contained in Weblogs (blogs) [10, 67], photos on sites such as Flickr [27], and videos on sites such as YouTube [68]. Web sites that center around user generated content are referred to in the media as Web 2.0 [58] to distinguish

them from conventional, single author, Web sites.

Adoption of Web 2.0 has been widespread, with users of all ages participating in posting as well as viewing content [51]. 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.

A true driving force of Web 2.0 has been multimedia user generated content. The success of multimedia user generated content is exemplified by the huge popularity of the video sharing Web site, YouTube. More recently, implementations of video sharing components have appeared in social networking Web sites [25, 55] to meet the demands of users who wish to share videos with their online friends. The popularity of multimedia user generated content combined with the availability of consumer broadband connections has the potential to severely strain the resources of both centralized servers and edge networks serving Web 2.0 users. A thorough understanding of the workloads of these Web sites is important for several reasons. First, understanding the workloads of Web 2.0 sites can highlight differences and similarities between the workloads of these sites and traditional Web sites. Second, an understanding of the workloads of Web 2.0 sites can facilitate the development of more current models of Web traffic. These models are useful for capacity planning at edge networks and central servers. Finally, knowledge of the workload characteristics of Web 2.0 sites can provide insights into which techniques will be most successful at dealing with the resource demands of these sites.

This thesis presents one of the first characterization studies of a Web 2.0 site. Specifically, this thesis focuses on the extremely popular video sharing Web site YouTube. While there are extensive studies of traditional Web workloads (e.g. [6, 7, 20, 46]), there have been no substantive studies of Web 2.0 workloads in the

literature. This thesis fills this void by characterizing YouTube usage in a campus network. The characterization performed in this thesis takes a multi-level approach, considering both aggregate properties of YouTube usage as well as user and session level properties.

The remainder of this chapter is structured as follows. The goals of this research are specified in Section 1.2 which is followed by a brief overview of the research contributions of the thesis in Section 1.3. This chapter concludes by outlining the organization of the remaining chapters in Section 1.4.

1.2 Objectives

The three main objectives of this thesis are as follows:

- develop a methodology to monitor all campus YouTube traffic for an extended period of time
- characterize the usage of YouTube on campus for an extended period of time
- gain insights into the resource demands of YouTube and draw implications for content providers, network administrators, and hardware manufacturers.

1.3 Contributions

This thesis has three main contributions [30,31]. First, a measurement methodology is presented that enabled campus YouTube usage to be monitored for an extended period of time. Second, campus YouTube traffic is characterized over a three month period. Finally, implications of the characterization results are provided.

There were many challenges faced when attempting to monitor campus YouTube usage. These challenges included the complex infrastructure used to deliver YouTube

content. This made monitoring *all* campus YouTube traffic difficult. The volume of data exchanged for campus YouTube traffic was another challenge that made monitoring campus YouTube traffic for an *extended period of time* difficult. In this thesis, a methodology for measuring campus YouTube usage is developed that addresses these challenges. This methodology involved determining the set of servers delivering YouTube content. Once the set of servers to monitor was determined, transactions between these servers and campus were summarized in real time to a log file. To minimize disk space requirements on the network monitor, the log file was compressed at the end of each day. Concurrent with the campus data collection, details of globally popular videos were gathered using methods provided by YouTube.

Using the aforementioned methodology, a characterization study of campus YouTube traffic was possible. YouTube characteristics are studied at different levels of aggregation. At the aggregate level, we studied usage patterns, file properties, referencing behavior, and transfer properties. At the user and session level, the amount of data transferred by users as well as several characteristics of user sessions are considered. Where possible, properties of YouTube traffic are compared to characteristics of traditional Web and streaming media workloads. Some key observations of the characterization are as follows:

- A small fraction of the requests to YouTube are for videos, but video downloads account for almost all of the bytes transferred.
- Video content has file sizes, transfer sizes, and transfer durations that are orders of magnitude larger than other content types.
- The concentration of references (defined in Section 2.2.3) is lower for campus YouTube content when compared with traditional Web workloads.

- Time between Web page requests ("think times") of YouTube users tend to be longer than those of other Web users. We attribute this to the time it takes users to watch a video and then move on to browse more pages.
- We observe session durations for YouTube users that are similar to those observed in previous studies.
- A large number of user sessions result from third party Web sites embedding YouTube content. As a consequence, we observe more sessions that transfer no videos than would be expected.

Results from our characterization are used to draw implications for content providers, network administrators, and hardware designers. For network administrators who desire to manage the bandwidth demands of YouTube traffic, we observe that there are many challenges associated with caching YouTube content. These challenges stem from the low concentration of references we observe as well as the large file sizes of video content. Content providers of sites like YouTube will also be challenged by the large file sizes of YouTube content. Specifically, larger disks will be required to store the wide array of content posted by Web 2.0 users. Hardware manufacturers are also impacted by the workload of sites like YouTube. Video content requires longer transfer durations than other types of Web content. This results in servers needing to maintain more concurrent connections. Multi-core processors that can handle this type of parallel workload will likely be desirable to providers of multimedia user generated content.

1.4 Thesis Organization

This chapter has presented the background and motivating factors of the thesis. Objectives and contributions of the thesis have also been summarized. The rest of

the thesis is laid out as follows. Chapter 2 presents relevant background material on networking and the Web. Related studies are discussed in Chapter 3. Chapter 4 develops the methodology used to measure YouTube usage from both the edge and global perspectives. Characteristics of campus and global YouTube data are discussed in Chapter 5. Campus YouTube data is analyzed at the user and session level in Chapter 6. Implications of our results and a comparison to related work are provided in Chapter 7. Finally, conclusions and future work are provided in Chapter 8.

Chapter 2

Background

This chapter provides relevant background on computer networking. The organization of the Internet architecture is discussed with specific attention to details that are relevant to studying YouTube. This is followed by an overview of the fundamentals of the Web. Subsequently, commonly used strategies for managing Web traffic are also presented. Background material on YouTube, the Web site studied in this thesis, is also presented. Finally, a primer on the statistical techniques used in this thesis is presented.

2.1 TCP/IP Network Architecture

The complex nature of the Internet necessitates the organization of its protocols. This organization comes in the form of a layered architecture of protocols referred to as the TCP/IP network architecture. Figure 2.1 shows the 5 layer structure of the TCP/IP network architecture. Protocols on the Internet are organized such that they each belong to one of these layers (with very few exceptions). The layered architecture enables a modular and flexible approach to be taken when designing network protocols [40]. For example, application layer protocols may choose to use either reliable data transfer provided by transmission control protocol (TCP) or the unreliable user datagram protocol (UDP) at the transport layer.

Protocols in the TCP/IP network architecture may be implemented either in software or on hardware such as network interface cards [40]. In general, protocols of the application layer, such as hypertext transfer protocol (HTTP) or file transfer protocol (FTP), are implemented in software. The same also holds for the transport

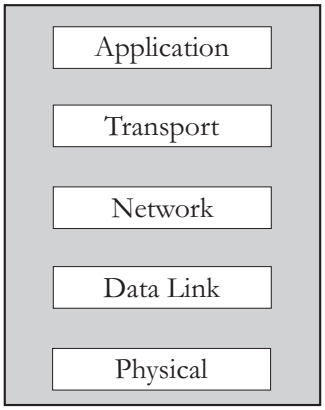


Figure 2.1: TCP/IP Network Architecture

layer protocols, UDP and TCP. At the network layer, Internet protocol (IP) is often implemented using a combination of software and hardware at the intermediate nodes along the network path. Link layer protocols and physical layer protocols are most often implemented in hardware at both intermediate nodes and end systems. It is also important to note that since the transport and application layers are concerned mainly with end-to-end reliable data delivery they are usually implemented only at end systems, whereas network layer, link layer, and physical layer protocols are also implemented on intermediate nodes in the Internet (e.g., routers). This section discusses the layers of the TCP/IP network architecture with an emphasis on the transport and application layers, which are most relevant to this thesis.

2.1.1 Application Layer

The application layer contains the protocols that most Internet users are familiar with. These include protocols that support electronic mail (e-mail), file transfers, peer-to-peer applications, instant messaging and multimedia streaming. These applications are implemented in software at the end-hosts in the network. Most relevant to the characterization in this thesis is the hypertext transfer protocol (HTTP) [26], which is a key component of the World Wide Web. HTTP is the protocol used for

communication between hosts on the Web. These hosts can include clients, servers, and proxies. HTTP and the Web are discussed in more detail in Section 2.2.

2.1.2 Transport Layer

Application layer protocols obtain service from transport layer protocols. Protocols at the transport layer are responsible for logical end-to-end transfer of data between processes on the source and destination hosts. In addition to transparency of data delivery, transport layer protocols may also provide services such as in order or reliable delivery of data. The dominant protocols at the transport layer are TCP and UDP. This section describes the method used by the transport layer to deliver data between processes and highlights key features of TCP and UDP.

Transfer of data between processes on the source and destination hosts is made possible by the notion of *flows* at the transport layer. Since a host may connect to multiple remote hosts, the notion of flows enables these connections to be logically separated. Each flow at the transport layer can be identified using a 5-tuple containing the Internet protocol (IP) address of the sender and receiver, the ports being used by the sender and receiver, as well as the transport protocol being used (primarily TCP or UDP). While the IP address of a host identifies the specific host that the data is being directed to, the port number enables separation between multiple applications on the same host. This process of delivering data to specific applications is sometimes referred to as multiplexing/demultiplexing of flows.

Transmission Control Protocol (TCP)

The most common protocol at the transport layer is TCP, which provides connection-oriented, in order, reliable delivery of data in addition to end-to-end transparency [39]. These services make TCP the transport protocol of choice for applications that cannot tolerate data loss, such as file transfer, e-mail, and the Web. While it provides

many services, TCP does not make any guarantees on the amount of time it will take to transfer data (referred to as *segments*), making it a poor choice for time sensitive applications.

To provide in order, reliable data transport TCP has a more complex structure than the alternative transport protocol, UDP. TCP uses a connection-oriented approach to data transfer and each packet that is sent using TCP is encapsulated in the TCP header. The TCP header fields are used to maintain state variables of the TCP connection. These include the current sequence and acknowledgment numbers, as well as several flags used for connection set-up, tear down, and data acknowledgment.

A TCP connection is set up using the `SYN` flag in the TCP header. The TCP client will send a TCP segment with the `SYN` flag set. The server responds to this initial message with a segment where the `SYN` and `ACK` flags are set. Once the client receives this segment (often referred to as the `SYNACK` segment), the client will respond with a packet with the `ACK` flag set. Sometimes this final `ACK` segment can also contain data to be transferred from the client to the server. This is called "piggy-backed" data.

Once the TCP connection has been opened data is sent between the server and the client. To ensure that data is transmitted reliably TCP makes use of sequence numbers and acknowledgment (`ACK`) segments. Each segment sent has an associated sequence number which cumulatively indicates that the Nth byte of the connection is contained within the segment. Once the data segment is received, the receiver will respond with an `ACK` segment where the `ACK` number is set to the sequence number of the received packet. The `ACK` segment lets the sender know that the data that was sent was successfully received. If an `ACK` segment is not received within a specific time period the sender will assume the data has been lost and resend it. If 3 duplicate ACK segments are received, corresponding to the data sent prior to the segment of interest, the sender will infer that the data has been lost and retransmit the segment.

During the TCP connection, there are four states that determine the sending rate of the connection, collectively referred to as *congestion control*. The sending rate of a TCP connection is reflected in the number of unacknowledged segments that will be permitted by the sender, denoted as *cwnd*. The first state is slow start which takes place as long as *cwnd* is below a predefined threshold, often denoted as *ssthresh*. During slow start, each acknowledgment causes *cwnd* to increase by 1 segment. Once *cwnd* reaches *ssthresh* the connection enters the congestion avoidance state where *cwnd* is increased cautiously so as not to cause undue congestion in the network. During congestion avoidance *cwnd* will increase by $1/cwnd$ for each acknowledgment that is received. If a segment is determined to be lost, the connection enters the fast retransmit and fast recovery stages. The implementation of fast recovery and fast retransmit depends on the variant of TCP in use [3, 29, 49]. However, most deal with loss by reducing the sending rate by half. This drastic reduction in sending rate combined with the progressive increase of *cwnd* during congestion avoidance is referred to as "additive increase, multiplicative decrease".

When all the data in a TCP connection has been sent the connection will be terminated using segments with the `FIN` flag set. First, the client will send a segment to the server with the `FIN` flag set. Once this segment has been received the server responds with an `ACK` segment and its own `FIN` segment. The client responds with an `ACK` segment once the server's `FIN` segment has been received. After sending its `ACK` segment the client then waits for a period of time before considering the connection closed. This timed waiting period enables the client to resend the final `ACK` segment if it is lost.

User Datagram Protocol (UDP)

TCP's congestion control often results in a time-varying transmission rate which may be undesirable for applications that have strict timing requirements that may

tolerate some loss. These applications generally perform better using UDP [60] at the transport layer. UDP provides minimal service at the transport layer [40]. Aside from the previously described multiplexing of flows, UDP simply passes data to be sent to the network layer where the Internet Protocol (IP) will deliver it from the source to the destination. It is important to note that service provided by IP is "best effort" and there are no guarantees on whether or not the data will be successfully delivered to the destination.

While UDP may sound quite primitive, many applications benefit from UDP's light-weight approach to transport service. For example, traditionally streaming media applications would use UDP to deliver data. Since there is no requirement for data to be acknowledged in UDP the sender may send data as fast as is possible given the network resources. Streaming media also tends to be tolerant of lost data and many techniques have been developed for dealing with lost packets in streaming media (e.g., [44]). While fast data transfer is desirable for applications like streaming media, UDP may adversely impact other flows using the same link. As a result, there have also been many studies that look at reducing these impacts and improving UDP's fairness (e.g., [28, 70]).

2.1.3 Network Layer

At the network layer, the Internet Protocol (IP) is responsible for delivering units of data referred to as *datagrams* from source to destination (denoted by IP addresses). The network layer is responsible for two main functions. These functions are forwarding (i.e., moving packets from the input to the output ports of a router) and routing (i.e., determining which path the datagram should take between the source and destination) [40]. The service model of IP as stated in the previous section is "best effort". This means that there are no guarantees on whether or not the datagram will be received at the destination, or how long it will take the datagram to arrive

at the destination. As we have discussed, this "best effort" service is abstracted from the application's view of the network by protocols like TCP that implement mechanisms to use IP's service to provide reliable data transfer.

Routing on the Internet is based on IP addresses. An IP address is a 32 bit number that distinguishes one host from another at the network layer. IP addresses are most commonly written as four 8-bit numbers separated by periods (e.g. `136.159.17.52`). Hosts on the Internet (with the exception of those located behind a network address translator (NAT)) require unique IP addresses to enable the network layer to route data to them [40]. To reduce IP address conflicts, the Classless Interdomain Routing (CIDR) address assignment strategy is used. Using this assignment strategy, Internet service providers (ISPs) are assigned "blocks" of address space referred to as subnets. A subnet is a set of IP addresses where the first x bits of the address are common. Subnets are denoted using the notation `a.b.c.d/x` where the first `x` bits of `a.b.c.d` are the prefix that is common among all addresses in the subnet. The remaining (32-x) bits in `a.b.c.d` are set to 0. Clients of ISPs may also have their own subnets. These clients will most commonly be allocated a block of addresses from their ISP's IP address block. For example, an ISP with the subnet `a.b.c.d/20` may allocate the subnet `a.b.c.d/22` to one of its clients. This thesis uses the concept of subnets to monitor traffic to and from YouTube on campus. While we only observe a subset of YouTube's address space, we are able to determine the address space allocated to YouTube by using the `whois` tools provided by regional Internet registries.

2.1.4 Data Link and Physical Layers

When delivering data from the source to destination, the network layer relies on link layer protocols to transfer data (referred to as *frames*) between intermediate nodes in the network. Different links may implement different link layer protocols [40]. Common link layer protocols include Ethernet and 802.11g. A network layer data-

gram may traverse many different links with many different link layer protocols on its path between the source and destination.

While the link layer is responsible for moving frames from one network component to another, the physical layer moves the individual bits contained in the frame between the components. Just as link layer protocols can vary depending on the link, physical layer protocols also vary depending on the transmission medium of the link [40].

2.2 World Wide Web

In the early days of the Internet many different networks existed. Many of these networks were deployed with the intent to connect researchers from various institutions to each other [40]. However, in the early 1990's the primary backbone network connecting these many networks (the NSFNET) began allowing the backbone to be used for commercial purposes [40]. By 1995, the NSFNET was decommissioned and the majority of backbone network traffic was carried by commercial ISPs. The popularity of the newly commercialized Internet was propelled by the development of the Web. Invented by Tim Berners-Lee between 1989 and 1991 [9], the Web provided the platform for many popular Internet services such as online shopping, news, and streaming multimedia [40]. This section presents a primer on the Web, its evolution, and strategies that have been proposed to manage the demands of Web traffic.

2.2.1 Hypertext Transfer Protocol

Hypertext transfer protocol (HTTP) is the application layer protocol used by the Web [26]. Implemented on both the Web client and Web server, HTTP dictates the format of messages sent between the client and server as well as the way in which the messages are exchanged. At the client, HTTP is implemented in a program

called a Web browser (e.g., Opera, Firefox, Internet Explorer). HTTP messages are primarily used by the client to request objects from the Web server. Objects on the Web can be equated with files on hard disks. Web pages may be made up of many objects embedded into a base `html` file which is also considered to be an object [40].

As stated in Section 2.1.3, hosts on the Internet are identified using IP addresses. While IP addresses are useful to lower layer protocols like IP, they generally do not appeal to human users. To make navigating the Web more intuitive, most Web servers also have a domain name. Translation between domain names and IP addresses is facilitated by the domain name service (DNS) [52, 53]. To enable addressing of individual objects on the Web, universal resource locator (URLs) are used. Each URL has two parts. The first part is the domain name which specifies the server that the object is located on. The second part of the URL is the path to the specific object [40]. For example, in the URL

`http://cpsc.ucalgary.ca/video.html`,

the domain name is `cpsc.ucalgary.ca` and the object's path is `/video.html`.

The HTTP message format provides some basic request types to be used by the HTTP client. The most common request types, often referred to as methods, are `GET` and `POST`. The `GET` method is used when the client is requesting a specific object from the Web server. Table 2.1 shows an example of an HTTP request using the `GET` method. In this example, the client is requesting the object `/index.html` from the server `www.myserver.com`. The final line in the table is the `User-Agent:` header field which is used to specify the user's browser. While the `GET` method is primarily used for retrieving data, the `POST` method is usually used to send data. The `POST` method is commonly used when users fill out online forms or post to message boards [26].

A sample HTTP response is shown in Table 2.2. The first field in the response is the status code which is preceded by the version of HTTP that is in use. In this

Table 2.1: Example of an HTTP Request

```
GET /index.html HTTP/1.1
Host:  www.myserver.com
User-Agent:  Opera/9.00
```

example, the status code is 200 which indicates that the request was successful. In general, status codes of the form 2xx indicate a successful response. Status codes of the form 3xx, however, indicate a redirection of the request. Most common are 303 "See Other" and 304 "Not Modified" which are commonly used for content distribution networks (CDNs) and caching, respectively. These concepts are elaborated upon in Section 2.2.3. Status codes of the form 4xx and 5xx are used if an error occurs either at the client or server. The `Date` and `Last-Modified` header fields indicate when the object was served as well as when the object was last modified, respectively. We use these later in our characterization to determine the age of content, and the time since the content was modified. Similar to the `User-Agent` field in the request header, the response header has the `Server` field to indicate what type of Web server is being used to serve the request. The final two fields in the header are the `Content-Length` and `Content-Type` fields, these indicate the size of the content (in bytes) and the type of content, respectively. We use these fields for many aspects of our characterization.

2.2.2 The Evolving Web

The development of the Web signaled a significant change in Internet usage. While originally used by academics to transfer files and send electronic data [40], the Web provided an application that sparked the interest of the general public. On-demand access to content was a very appealing aspect of the Internet [40]. With the click

Table 2.2: Example of an HTTP Response

```
HTTP/1.1 200 OK
Date:  Wed, 05 Dec 2007 20:30:43 GMT
Last-Modified:  Mon, 16 May 2005 15:59:15 GMT
Server:  Apache/2
Content-Length:  1545
Content-Type:  text/html
```

of a mouse Web users were able to gain access to a wide range of information. The concept of on-demand access to content led to widespread adoption of the Web and related Internet applications (e.g. e-mail, instant messaging). This widespread adoption led to the traditional Web publishing paradigm, where content is provided by the owner of the Web site and consumed by a large population of Web users.

As Web technologies developed, more complex interaction with Web sites became possible. The technological driving force behind the interactive features available on the Web today is a combination of Javascript and XML, commonly referred to as Asynchronous Javascript and XML (AJAX). As users have become able to participate and contribute to Web sites, a new Web publishing paradigm has emerged. Now, instead of Web sites containing content posted only by the Web site owner, Web users are also able to view and post content on many different Web sites that make use of these new Web technologies. Content posted by Web users is commonly referred to as *user generated content* and comes in a variety of forms including, textual information contained in Weblogs (blogs) [10, 67], photos on sites such as Flickr [27], and videos on sites such as YouTube [68].

Technology used by this next generation of Web sites (commonly referred to as Web 2.0 sites), has signaled a shift not just in content production, but in computing. Web applications are now performing tasks that were previously only possible using

desktop applications. Google's suite of online software tools such as spreadsheets and calendars exemplify this new trend of providing software on the Web instead of as an installed application [32].

Web-based applications combined with the popularity of multimedia user generated content place many demands on network resources. These demands impact edge network administrators who must deal with the bandwidth requirements of users consuming the variety of content available on Web 2.0 sites. Server administrators that provide highly interactive Web content are also impacted by the resource demands of these new applications. This thesis seeks to begin to understand the demands of Web 2.0 sites by characterizing the demands of one multimedia Web 2.0 site, YouTube.

2.2.3 Strategies for Managing Web Traffic

As the Web began to experience widespread adoption, its mass popularity necessitated the development of strategies to help manage the demands of Web users on the Internet. Two common approaches for managing Web traffic are caching and content distribution networks (CDNs). These approaches are similar in that they introduce hosts that act as intermediates between the client and the primary server hosting the Web content.

Caching is a technique that has been previously applied to memory and file referencing in operating systems [13, 47]. As the Web grew in popularity, the idea of caching was applied to Web content. Caching Web content has many advantages. Caches are generally located closer to the Web client resulting in lower latency when servicing client requests. Caching is also advantageous to the primary server hosting Web content as requests that would normally have to be serviced by the Web server can be served by the cache instead.

Two principles that caching relies on are concentration and locality. Concentra-

tion refers to the distribution of requests across the set of stored content. For caching to perform well requests should be concentrated on a small set of content. This is sometimes called the "Pareto" principle or the "90-10 rule". These terms apply when a small set of the available content (10-20%) accounts for a large fraction of the requests to a server (80-90%). This principle has been observed in empirical analysis of Web and streaming media traffic [7, 62, 69]. Another method that is commonly used to evaluate the concentration of requests is Zipf analysis [72]. Zipf's law relates the number of requests for content with its rank (in terms of requests) according to the following formula:

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

where F is the number of requests for an object and R is the relative rank of the object. A stream of references is said to follow Zipf's law if β is close to 1 when the objects are ranked in order of descending popularity. Otherwise, the distribution of references is said to be "Zipf-like". As the value of β increases, so does the concentration of requests.

Locality is another important principle that caching relies on. In this thesis, we focus on temporal locality which 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 [23]. Similarly, locality has also been found to occur in Web server and proxy document reference streams [7, 45, 46].

Content distribution networks (CDNs) replicate content stored on servers. These replications are placed in edge networks that are physically closer to the clients requesting the content [40]. There are many benefits to CDNs. Web site users benefit from lower latency in their requests for the Web content. The lower tier Internet service providers (ISPs) where the content replications are located also benefit from

reduced peering costs with neighboring ISPs (as a result of more traffic remaining within their network). The Web site using the CDN benefits from reduced load on central servers as well as potentially higher profits because of improved quality of service for their clients. A limiting factor of CDNs is their cost. As a result, most Web sites prefer to serve content themselves to save on the cost of paying for a CDN.

As the Web continues to grow and evolve it is important to evaluate the methods used to manage the demands of Web traffic. This is especially important with the widespread popularity of user generated content which places significant demand on network resources. This thesis provides some insights into the applicability of these traditional methods for managing Web traffic for a popular, multimedia-based, Web 2.0 site.

2.3 YouTube

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 [65]. 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 [63]. 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 plug-ins 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

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

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).

2.4 Statistical Background

When analyzing network measurements it is common to treat the data as if it was produced by a random process [21]. Thus, probability is a useful tool when considering these datasets. Three methods for considering probability distributions are used to analyze data in this thesis: probability density function (PDF), cumulative density function (CDF), and complementary cumulative density function (CCDF). These are discussed in turn.

The PDF of a random variable X is defined as the probability that X takes on a specific value x. In network measurement X refers to the the observed data points which are assumed to be caused by a random variable. The PDF is commonly denoted as $P(x) = P[X = x]$. The PDF can be graphed by putting the potential values of X along the x axis and the number of times x occurs divided by the total number of observations on the y axis (e.g., Figure 5.7). Since data can often takes on numerous values with fluctuating frequencies binning is often required in order to create a PDF that shows the overall shape of the distribution. Binning involves grouping values into intervals (e.g., 1 minute intervals in Figure 5.7).

The CDF considers the probability that X falls below a specific value x, denoted as $F(x) = P[X < x]$. The CDF enables the the distribution to be considered in detail without binning which removes some detail. In this thesis, CDFs are plotted with $P[X < x]$ along the y axis and the values of x along the x axis (e.g., Figure 5.8).

While CDFs provide a details view of the distribution of a random variable,

CCDFs are often used to obtain a better view of the tail of the distribution (e.g., Figure 5.20). The CCDF is denoted as $(1 - F(x)) = P[X > x]$. The tail of distributions is of particular interest in network measurement where the tail of distributions tends to decline much more slowly than the tail of distributions commonly considered in probability theory [21]. This has the consequence that extremely large values are often observed with non-negligible frequency in network data. Two concepts related to this slow decline in the tail values of distributions are *long tails* and *heavy tails*.

A distribution is said to have a long tail if its tail declines subexponentially [21]. More formally

$$(1 - F(x))e^{\lambda x} \longrightarrow \infty \text{ as } x \longrightarrow \infty \text{ for all } \lambda > 0.$$

Long tailed distributions will have extremely high, or even infinity variance.

A special case of the long tail distribution that is often observed in network measurements is the heavy tailed distribution (e.g., Pareto distribution). In the case of the heavy tail distribution the tail approaches to a power law distribution as it declines [21]. Formally, a heavy tail distribution is observed when

$$(1 - F(x)) \sim x^{-\alpha} \quad 0 < \alpha \leq 2$$

This distribution has extremely high variance and when $\alpha \leq 1$ the distribution will have an infinite mean [21]. Heavy tail distributions have been linked to self-similarity, a distinguishing feature of network traffic [21]. Self-similarity refers to correlations between a process and itself when observed on varying time scales.

2.5 Summary

This chapter gave an introduction to the concepts that will be used in this thesis. The Internet architecture and its layers were discussed with an emphasis on the

application and transport layers. At the application layer, key aspects of the Web including HTTP, the Web's evolution and methods that have been developed for handling Web traffic were presented. Details of YouTube, the Web site characterized in this thesis and relevant statistical tools were discussed.

Chapter 3

Related Work

Characterizing workloads of Web sites is not a new avenue of study. There have been many studies that characterize the workloads of traditional Web sites. The workloads of Web sites that center around streaming media content have also been considered in the literature. These previous studies of conventional Web sites and streaming media sites are summarized in Sections 3.1 and 3.2, respectively.

3.1 Web Workload Characterization

There are many related studies on Web workloads [5–8,22,24,33,46,48,50,57]. These studies can be divided into two categories; those that consider the workload in terms of general workload characteristics [6,7,24,33,46] and those that consider the workload in terms of user sessions [5,8,22,48,50,57]. Papers that fall into these categories are considered in turn in the following sections.

3.1.1 General Web Workload Characterization

This section discusses previous work that focuses on general characteristics of Web workloads. These characteristics can include file sizes, request sizes, and document popularity. General characteristics of Web traffic are important to study because they can give insights into the amount of stress placed on network resources by Web workloads, both at the server and at the edge network. These insights are useful for developing strategies to handle the demands of Web workloads.

Invariants that are applicable to Web server traffic are sought out by Arlitt and Williamson [7]. In this study, the authors consider the workloads of six different

Web servers with the goal of finding properties that will hold for Web traffic in general. Prior to this work, most studies were based on a single data set which was often limited by where it was collected (e.g., an academic or enterprise setting). The authors are able to identify 10 properties of Web traffic that hold across all of the servers they consider. These properties include the fact that the majority of content transferred is image and HTML data. They also observe consistent heavy tailed distributions of file and transfer sizes as well as very concentrated referencing behavior with 10% of the files accessed accounting for 90% of the bytes transferred by the server. This very concentrated referencing behavior is consistent with the Pareto principle. The authors revisit these results several years later at the three academic sites studied [66]. They find that despite a 30-fold increase in traffic the 10 original invariants still hold.

Gribble and Brewer study characteristics of Web usage by users of a dial up Internet service provided to university users [33]. Within their traces the authors observe strong diurnal effects as users are most active during the day time and evening hours. They also observe file sizes that tend to be quite small, typically less than 10 KB. Using a simulated cache they make observations about the potential for caching to benefit the population of users observed in their traces. Through these simulations the authors find that as the user population increases, caching efficiency also increases as long as the cache is large enough to support the working set of the larger population.

Workload characteristics of the extremely busy Web server for the 1998 FIFA[1] World Cup are considered by Arlitt and Jin [6]. In that study, the authors consider the workload of the Web site over a period of 3 months. Since the World Cup event was very popular, the server studied was heavily loaded, especially at times when the soccer games were taking place. Similar to the aforementioned work by Arlitt

[1]Federation Internationale de Football Association

and Williamson, it is observed that the majority of the content transferred is images and HTML. They also notice a large amount of cache control traffic, evidenced by many HTTP transactions with status 304. These transactions were especially evident at times when the server was very busy, this indicates that cache consistency mechanisms (at the time of the study) were not effective at stopping the problem of "flash crowds" at Web servers.

Mahanti *et al.* characterize the workloads of Web proxies from three different levels of a caching hierarchy [46]. They consider several characteristics of Web workloads in their proxy datasets. Specifically, document types, transfer sizes and referencing behavior are considered. Similar to Arlitt and Williamson, the authors find that images are the most commonly transferred document type, followed by HTML content. They also observe that files transferred at the proxies tend to be small with mean and median being under 20 KB. It is also observed that the concentration of requests is lower at Web proxies than at Web servers. The authors reason that this is because users of a Web server are limited to requesting content available on the server, whereas at a Web proxy the user may request content from a larger set of Web sites. This would result in the requests being distributed across a much larger set of content, thus reducing the concentration of requests.

3.1.2 Session Level Characterization

A user session is defined by Menasce *et al.* as a series of requests issued by a user to a Web site in a single visit to the site [50]. Understanding user sessions is an important step in the development of Web workload generators. In this section, previous studies that consider user session characteristics in Web workloads are considered.

Barford and Crovella develop a Web workload generator based on the concept of modeling "user equivalents" [8]. To model user equivalents, the authors consider several characteristics of Web traffic. Specifically, they consider the distributions

of file and request sizes, popularity of the files and inter-transaction times of the users in their traces. When considering inter-transaction times the authors separate automated requests of content embedded in Web pages from user generated requests using the notion of active and inactive "OFF" times. Active "OFF" times are meant to represent delay induced by automated requests, whereas inactive "OFF" times are used to represent user think-times. A threshold value is used to distinguish active "OFF" times from inactive "OFF" times. An "OFF" time is considered active if its duration is less than a threshold value. An "OFF" time with a duration greater than the threshold is considered inactive (ie., it is considered to be a result of the user think time between object requests).

Characteristics of user sessions of the 1998 FIFA World Cup Web site are studied by Arlitt [5]. That study focused on user session behavior to improve end user experience. A method of improving user experience that is considered by Arlitt is increasing the number of concurrent users by improving persistent connection management in HTTP 1.1 servers. Specifically, it is found that while a simple timeout based approach may reduce the total number of active TCP sessions a server must maintain, an adaptive timeout approach would minimize the amount of redundant timeouts that would occur for sessions that only have a single request.

User sessions that use multiple applications on a broadband network are characterized by Marques *et al.* [48]. They consider both residential and small business users. They observe interarrival times between sessions that are exponentially distributed for both categories of users they consider. As expected, a difference they observe between the two user groups is the diurnal patterns of small business users are more pronounced than for the residential users.

3.2 Multimedia Workload Characterization

YouTube is currently the largest video sharing Web site on the Internet [43]. The repository of content on YouTube is also growing at a rapid pace, with 65,000 video uploads per day [65]. Understanding the characteristics of this media workload and how it compares to traditional multimedia workloads is important when determining which delivery strategies will be most efficient for delivering YouTube content. The workloads of online multimedia services have been well studied in related literature [1, 2,4,17–19,34,35,37,42,62,69]. This section summarizes related studies of multimedia workloads that are most relevant to this thesis.

A popular delivery model for large media files in recent years has been the peer-to-peer delivery model. Gummadi *et al.* characterize the workload of the KaZaA peer-to-peer system [34]. In their study, the authors consider the referencing behavior of KaZaA clients. They find that in contrast to Web content (where referencing behavior follows a Zipf distribution), referencing in the KaZaA system is not well modeled using a Zipf distribution. They observe that the most popular content is not as popular as would be expected if the referencing behavior followed a Zipf distribution. The authors attribute this non-Zipf behavior to "fetch-at-most-once" behavior. They reason that in a peer-to-peer system, once the user has downloaded a file, the user will not need to download it again. This is in contrast to the Web where Web pages are updated, causing users to revisit them to view these updates.

Live streaming workloads are considered by Sripanidkulchai *et al.* [62]. This study uses data from a large number of streaming servers operated by Akamai Technologies, a large content delivery network (CDN). In their study, the authors observe distinct time of day and day of week effects when they consider users subscribing to non-stop streaming radio. It is found that during the day, more users are subscribed to the live stream. Also, on weekdays the number of users subscribed to the stream is higher

than on weekends. When considering the popularity of the live streaming content, that study finds that requests for the live streams are well modeled using a 2-mode Zipf distribution. The 2-mode Zipf distribution they observe arises from fewer less popular streams than would be predicted by a single mode Zipf distribution. Since their data comes from a CDN this result is not surprising. Content publishers are unlikely to pay for a CDN to deliver unpopular content that the publisher may be able to deliver themselves.

Li *et al.* consider characteristics of streaming media content on the Web using a Web crawler [42]. The authors find that the median duration of the streaming content is 3 minutes. They also observe that the distribution of media durations may be long tailed. This observation indicates that the transfer times for these media objects may also be long tailed, which can result in the Internet traffic associated with streaming applications being self similar. When considering the bit rate of the streaming content the authors observe a median bit rate of 200 Kbps. It is also noted in that study that the most prevalent formats of media content are proprietary with the dominant commercial products being RealPlayer and Windows Media Player. This is interesting to note, as the study by Li *et al.* was done prior to YouTube and the popularity of using Flash to deliver video content.

Workload characteristics of a large scale video-on-demand (VoD) system are considered by Yu *et al.* [69]. Similar to the work by Sripanidkulchai *et al.* [62], Yu *et al.* observe request patterns that are impacted by the time of day. Since the VoD service studied is primarily for entertainment, Yu *et al.* observe more activity in the evening hours than during the work day. The authors also observe that users of the VoD system tend to be impatient with 86% of the sessions being terminated before video playout is 85% complete. Of these incomplete transactions, 52% of them are terminated within the first 10 minutes. The authors conclude that despite the availability of program guides users like to watch the first part of the content to determine if

it will interest them. When considering the reference behavior of users of the VoD system, the authors find that the referencing behavior follows a Zipf distribution, with the exception of the least popular content.

More recently, Guo *et al.* studied the referencing behavior of users across a wide variety of Internet media systems [35]. This study considers a wide range of media workloads including Web, VoD, peer-to-peer and live streaming environments. Across all of their datasets the authors perform analysis and show that the referencing behavior of media objects follows a stretched exponential, rather than a Zipf distribution.

3.3 Summary

This chapter summarized previous studies that are most relevant to this thesis. These previous studies fall into two main categories: Web workload characterizations, and multimedia workload characterizations. YouTube uses a Web framework to deliver streaming multimedia content to its large user base from its fast growing repository. This makes understanding previous work on Web and multimedia workloads important when considering the similarities and differences between the workload of YouTube and these previously well studied workloads. An understanding of how YouTube compares to these other types of workloads can also help provide insights into strategies that will be most effective to handle the resource demands of YouTube usage.

Chapter 4

Methodology

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. This chapter presents a multilevel approach to capturing YouTube traffic and understanding its workload characteristics. First, YouTube usage is monitored on the University of Calgary campus network. By considering local YouTube usage it is possible to gain insight into how YouTube may be used by clients of other large edge networks. Section 4.1 describes the local data collection methodology. Second, statistics were collected on the most popular videos on the YouTube site. This global data collection is described in Section 4.2. By keeping statistics of both local and global YouTube usage it is possible 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

To gather information on the impacts of YouTube usage on network resources this thesis takes an edge network point of view. The edge network considered is the University of Calgary campus network which consists of approximately 28,000 students and 5,300 faculty and staff [56]. Goals and challenges faced when measuring the university network as well as our measurement methodology are discussed in this section.

4.1.1 Goals and Challenges

When monitoring campus YouTube usage there were three main goals of the data collection. First, the data collection methodology needed to be able to capture all usage of YouTube on campus. Second, the ever changing nature of YouTube motivated measurement of campus YouTube usage for an extended period of time. Finally, the data collection of campus YouTube usage needed to maintain user privacy to an acceptable degree.

While the goals set forth for the measurement seemed straightforward, there were many challenges that needed to be overcome. The first challenge faced when attempting to monitor campus YouTube traffic was the extreme popularity of YouTube. Due to the large number of users that view and post content on YouTube each day, YouTube has developed a complex delivery infrastructure. This infrastructure includes many servers within YouTube's network as well as servers belonging to a content delivery network (CDN). The network monitor used in this thesis posed the second challenge to monitoring campus YouTube traffic. The monitor was purchased in spring 2003 when the campus Internet link was only 12 Mb/s; it has two Pentium III 1.4 GHz processors, 2 GB of RAM and two 70 GB hard drives. While the monitor was sufficient at the time it was purchased, it does not have the processing power or disk space to keep up with the 300 Mb/s link that currently connects the campus to the Internet. The high speed of the link connecting the University of Calgary campus to the Internet was another challenge faced when performing data collection on campus YouTube usage. Prior to our study, the university upgraded the Internet link from a 100 Mb/s to a 300 Mb/s full duplex link. Soon after the upgrade, campus users began taking advantage of the newly available bandwidth. Figure 4.1 shows the aggregate bandwidth (inbound + outbound) consumed on our campus Internet link during the collection period. This increased use of the campus

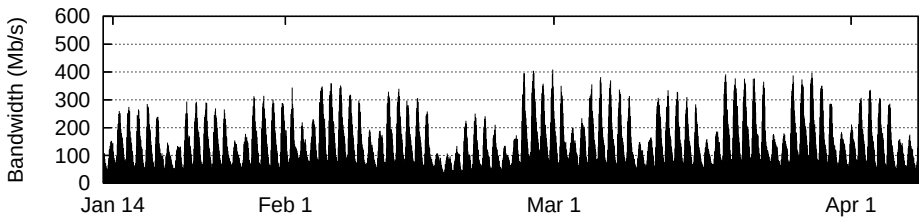


Figure 4.1: Aggregate Campus Internet Bandwidth During Collection Period

Internet link places a great deal of pressure on our aging monitor.

4.1.2 Our Measurement Solution

To address the aforementioned challenges, the following methodology was used. First, the set of networks providing YouTube content was determined. This involved using `tcpdump` [64] while browsing YouTube to find the IP addresses of servers delivering YouTube content. Once the set of servers was collected, a `whois` service [1] was used to determine the network each IP address belonged to. As expected, most of the IP addresses were within YouTube's networks. In addition to YouTube's networks, we also observed the Limelight content delivery network serving YouTube content to campus users.

Once the set of networks to monitor was determined, `bro` [12] was used to monitor HTTP transactions between these networks and campus users. However, it was quickly noticed that Limelight was being used to deliver more than just YouTube content. Specifically, Limelight was being used to deliver other popular Web sites such as Facebook and MSNBC. As a result of this, a second step of filtering had to be performed to determine which of the Limelight traffic was YouTube related. Since the same IP address could be used to deliver content for multiple sites, filtering based on IP address alone was not sufficient. To resolve this issue, the host name of the server was considered. It was noticed that the fully qualified domain name of

[1] `www.arin.net`

servers delivering YouTube content from the Limelight content distribution network contained the string "youtube". Thus, to determine if traffic was YouTube related, first the network the remote IP address belonged to was determined. If the network was one of YouTube's subnets the transaction was considered to be YouTube traffic. If the network was one of Limelight's subnets, then the `Host` field of the server would be checked to see if it contained the string "youtube". If the `Host` field contained the string "youtube" then the transaction was determined to be a part of campus YouTube traffic.

For each transaction that was determined to be YouTube traffic, `bro` extracted several summary statistics in real-time. These included data from both the transport and application layer headers, however, this thesis focuses mainly on the application layer data. The statistics collected by `bro` include application layer statistics such as, HTTP status code, method, content type, content length, date, and visitor ID (a cookie value). At the transport layer, duration and start/end sequence numbers of the TCP connection are collected. Many of the statistics were collected directly from header fields, however, the visitor ID is mapped to a unique integer value to protect user privacy. Once the transaction terminated, these statistics were written to a log file on the monitor machine. Each day at 4:00 am `bro` was restarted and the log from the previous 24 hours was compressed. This enabled the logs to be moved from the monitor machine without disrupting trace collection. Since the mapping of the visitor ID field to an integer was not stored to disk, the visitor ID is only valid within a single log file (i.e. for a 24 hour period). While this limited some of the analyses of user behavior that could be performed, it provided a higher level of user privacy.

After some initial experiments with `bro` on the network monitor, a "status" flag was added to each transaction summary. This flag indicates the status of each transaction, which falls into one of four categories:

Table 4.1: Breakdown of Transactions

Category	Transactions	% of Total	Video transactions	% of Video
Complete	22,403,657	90.82	154,294	24.66
Interrupted	462,903	1.88	151,687	24.25
Gap	383,878	1.56	319,612	51.09
Failure	1,418,178	5.75	-	-
Total	24,668,616	100.01	625,593	100.00

- **Complete:** the entire transaction was successfully parsed.
- **Interrupted:** the TCP connection was reset before it completed.
- **Gap:** the monitor missed a packet, and `bro` was unable to parse the transaction for an unknown reason.
- **Failure:** `bro` was unable to parse the transaction for an unknown reason.

Table 4.1 summarizes the prevalence of each of the transaction categories [2]. As expected, most of the transactions belong to the "Complete" category. Approximately, 6% of the transactions fall into the "Failure" category. No information from HTTP headers is available for transactions in this category. The two most likely causes of failed transactions are: the network monitor dropping a packet in the connection before the HTTP headers were parsed; or the TCP connection was not established in an expected manner, so the script was unable to handle it properly. Since neither of these issues are related to the type of object being transferred it is unlikely that a disproportionate fraction of failed transactions were for a particular content type. A count of the number of failed transactions is maintained to ensure that the majority of YouTube traffic is being captured. The remainder of this thesis does not consider transactions in the "Failure" category.

The prevalence of each of the transaction categories for video content is also summarized in Table 4.1. Only 24.66% of video transactions are completed. This

[2]The total is 100.01% due to rounding error

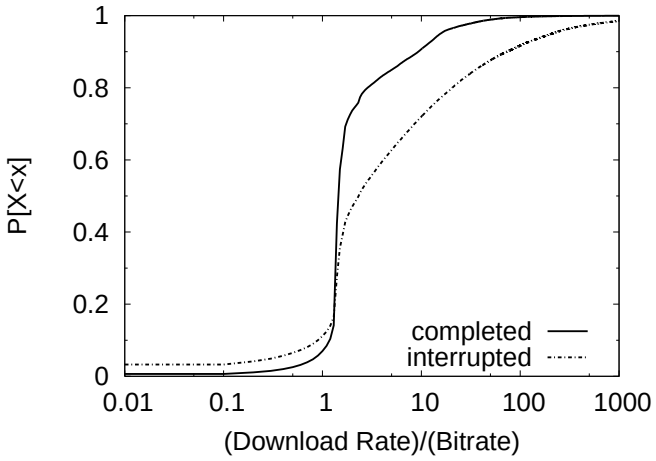


Figure 4.2: CDF of Ratio of Download Rate to Bit Rate for Video Transfers (Campus)

can be attributed to the large number of video transactions that fall into the "Gap" category. It can be seen in Table 4.1 that almost all the gapped transfers are for video content. This is due to the network monitor being unable to keep up with the network load on campus. Video transfers are impacted the most because they achieve much higher download rates than most other (smaller) transactions, thus placing a higher load on our monitor. When a "gap" is observed in a connection the transaction summary is output immediately without waiting for the transfer to terminate. This results in some of the collected statistics being invalid for these transactions. Namely, the transfer duration, and starting and ending sequence numbers are impacted by "gaps". However, many of the statistics are obtained from HTTP headers which are seen in the first few packets exchanged in a transaction. As a result, we are able to apply most of our analyses to "gapped" transaction as well.

Approximately 24% of video transactions fall into the "interrupted" category. Transactions in the "interrupted" category are those that were terminated before the transfer of data had completed. Since interruptions occur after the exchange of HTTP headers, these transactions can also be used in the analyses. There are two

primary reasons why a video download may fall into this category: *poor performance* (i.e., slow download rate); or *poor content quality* (e.g., the viewer does not find the content interesting). Figure 4.2 demonstrates this quantitatively. For example, approximately 10% of the interrupted transactions had a slower download speed than the encoded bit rate for the video (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 it can be conjectured that the users simply found the content uninteresting, and aborted the transfer.

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 method for examining global YouTube file characteristics. There are several reasons that crawling is not feasible. YouTube's video repository is considered to be the largest on the Internet, and is growing at an estimated 65,000 videos each day [43]. The large size of YouTube combined with its continual growth make random sampling very difficult without the use of specialized sampling methods [41]. YouTube's user policy also places limits on the types of automated browsing that can be undertaken.[3] Given the size of YouTube's repository, obtaining a sufficiently large sample of content would take an impractical amount of time if the user policy is to be respected.

To avoid the aforementioned issues with crawling YouTube, this thesis focuses 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. This choice was also motivated, in part,

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

by empirical evidence of the Pareto principle (or the so called "80-20" or "90-10" rule) in the file referencing behavior 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.

A two step approach is utilized to collect data on the top 100 videos on YouTube. First, each day the pages listing the most viewed videos of the day, week, month and all time are retrieved. These pages provide the video identifiers of the top 100 videos in each of the time frames. 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 the identifiers are gathered 20 page loads are performed (five for each of the four time frames).

The second step of data collection involves using APIs that are provided by YouTube for developers.[4] 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 created for this thesis) the function returns an XML object containing 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 the 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 during the measurement period. Thus, these requests are not filtered from the dataset.

[4] `http://www.youtube.com/dev`

4.3 Summary

This chapter presented the methodology used for collecting the data that will be analyzed in subsequent chapters. The data used in this thesis comes from two main data sets. The first data set is the campus data set, which was collected over a three month period on the University of Calgary network. The second data set is collected from the list of globally popular videos on YouTube, over the same time frame as the first data set. Challenges associated with measuring YouTube usage both locally and globally as well as the strategies used to address them are discussed.

Chapter 5

Characterization of YouTube Traffic

Characteristics of campus YouTube traffic at an aggregate level are considered in this section. Trends in YouTube traffic during the measurement period are discussed in Section 5.1. Properties of files transferred by campus YouTube users are presented in Section 5.2. File referencing behavior and locality characteristics are considered in Sections 5.3 and 5.4, respectively. Section 5.5 discusses properties of the file transfers such as transfer size and duration.

5.1 Preliminary Analysis

This section presents high-level characteristics of the dataset used in this thesis. Summary statistics of the data collected from the university campus network are presented in Section 5.1.1. Section 5.1.2 describes characteristics observed in the YouTube traffic on the university network over the course of the measurement period. Section 5.1.3 discusses summary statistics from the global YouTube data.

5.1.1 Local YouTube Summary Statistics

YouTube traffic to and from the University of Calgary campus network was monitored for 85 consecutive days, starting on January 14, 2007 and ending on April 8, 2007. Table 5.1 presents summary statistics for this traffic. The monitoring period subsumes important transition 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 the monitoring period is long enough to capture longitudinal changes in the characteristics of YouTube traffic.

Table 5.1: 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

Table 5.2: Breakdown of HTTP Request Methods

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

In total, 23,250,438 valid (i.e., non-failed) HTTP transactions (i.e., request/response pairs) were recorded. These transactions account for approximately 6.54 TB of data transferred on the University of Calgary network. 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 [59]. It is 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 5.2 presents a breakdown of the HTTP request methods seen in the YouTube campus trace. This analysis provides insights into the activity of YouTube users on our campus network. As expected, HTTP GET requests constitute the majority of requests. This indicates that almost all requests are for fetching content from

Table 5.3: 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

YouTube. There are also 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.

The content-type field of the HTTP POST messages is analyzed 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. Only a small number of POSTs are video upload attempts (133) over the three month collection period. The upload/download behaviors observed on the university network are likely 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 [43]. Clearly, most of the users are consumers of content and only a handful of the users are content producers, just as on our campus.

HTTP response codes provide additional insights into YouTube's workload. The breakdown of response codes is shown in Table 5.3. 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. Approximately 1% of the HTTP 304's are found to be for Flash Video, which suggests some users were re-watching selected videos. HTTP response codes 200, 206, and 304 make up 94% of the responses seen in our campus YouTube traffic. We also find 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 5.4. The results show that images (e.g., image/jpeg, image/png, image/gif) and text (e.g., text/html, text/css, text/xml) make up 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.

Table 5.4 presents characteristics of the unique files that were downloaded from

Table 5.4: 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

YouTube in the row labeled File Size. 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.

Table 5.4 also shows the transfer size statistics in the rows labeled transfer size. For Images and Videos, the transfer size statistics are quite similar to the file size statistics. However, Text and Application show some differences between the transfer sizes and file sizes observed. For Text, the transfers are mostly for a few smaller objects resulting in a lower average transfer size. The transfer size for application data is larger than the the file size. This is caused by many transfers of a few larger application files. Additional information is available in Section 5.2.1 and Section 5.5.1.

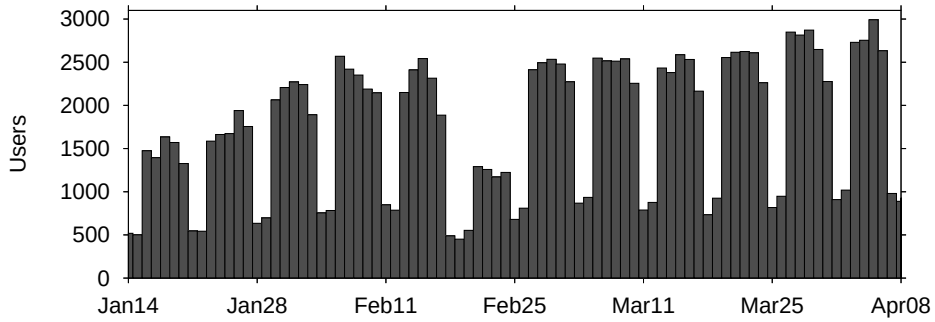


Figure 5.1: Unique Users per Day

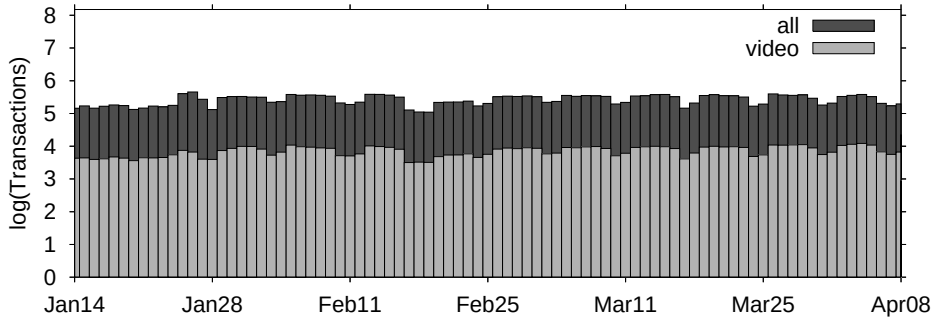


Figure 5.2: Requests per Day

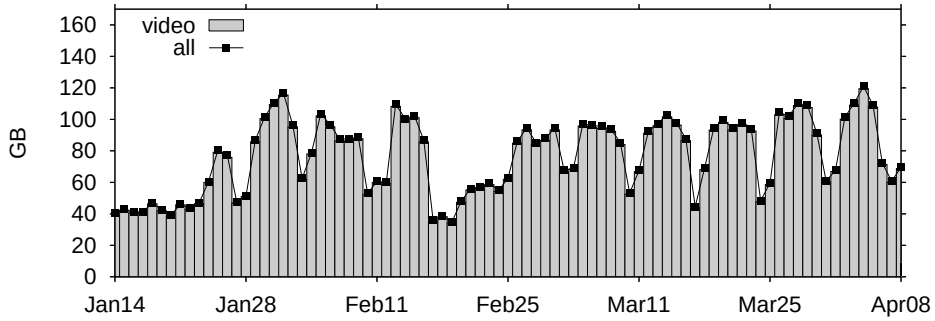


Figure 5.3: Bytes per Day

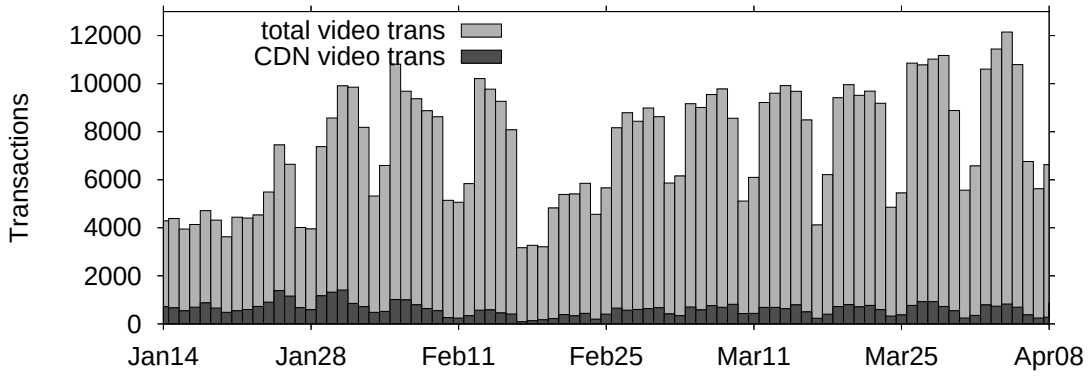


Figure 5.4: Video Requests Served by YouTube/CDN

5.1.2 Local YouTube Usage Characteristics

Figures 5.1, 5.2, and 5.3 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, and increases slowly thereafter, 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 probable 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 [54]. Simultaneously, a high profile viral marketing campaign on YouTube raised awareness of the site [14]. Traffic decreases in mid-February as a result of reading break, when many students leave campus.

Figure 5.2 shows that the number of requests for video is approximately two

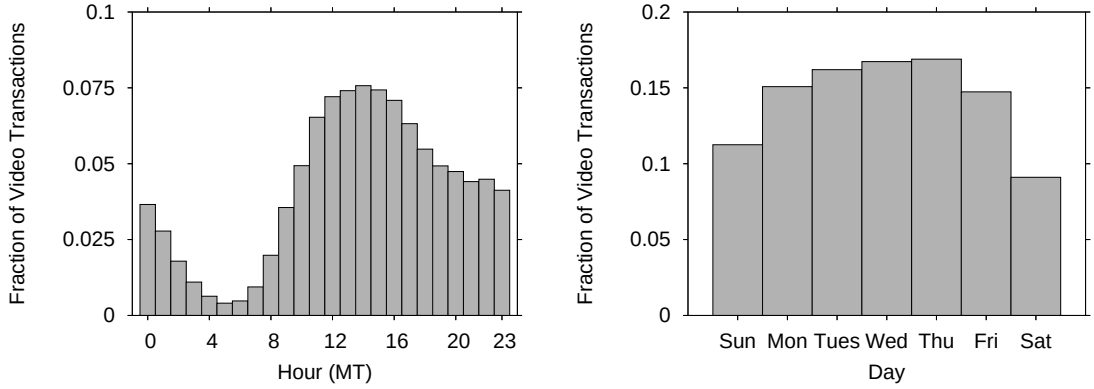


Figure 5.5: YouTube Traffic Patterns: (a) by time of day; (b) by day of week

orders of magnitude less than the total number of requests owing to YouTube use; however, as shown in Figure 5.3 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 5.4 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 5.4, 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 5.5(a) shows the fraction of total video requests seen at a particular time of day, while Figure 5.5(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

Table 5.5: 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

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.

5.1.3 Global YouTube Characteristics

Table 5.5 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.

5.2 File Properties

This section presents a characterization of the files observed during our measurement period. Characteristics that are considered include file sizes, video duration, bit rate, age of content, video ratings and video categories. An understanding of the properties of files transferred by a site like YouTube has many potential applications. Hardware manufacturers who will be designing the servers that will be storing and delivering Web content need to take into account properties of files to be stored. File sizes also have implications for cache management policies where caching large files may displace many smaller files in the cache.

5.2.1 File Size

Unique file sizes for video and non-video content types are considered in Figure 5.6. 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

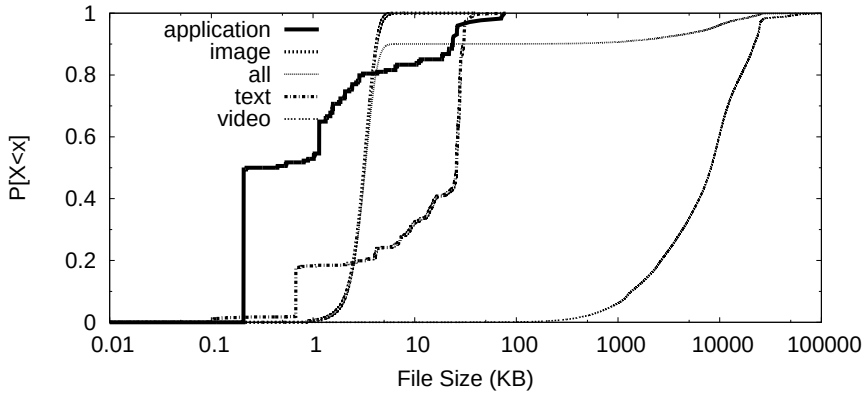


Figure 5.6: CDF of Unique File Sizes (Campus)

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.[1] 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.

5.2.2 Video Duration

In this section we analyze the duration of video files seen in our traces and make comparisons with durations for the globally popular videos that 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

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

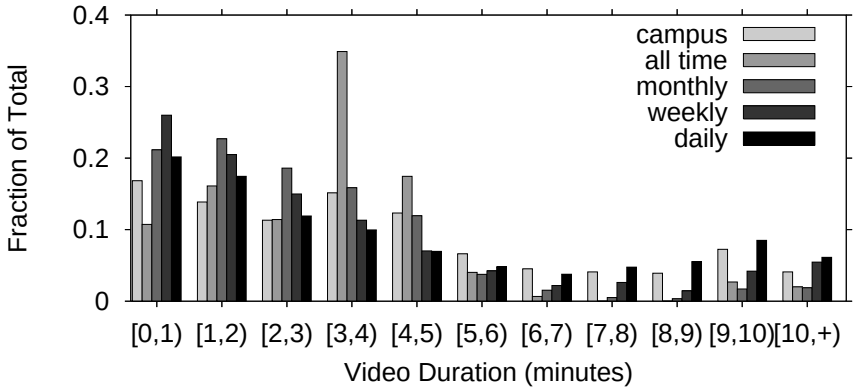


Figure 5.7: Histogram of Video Durations

this information for data collected locally. Figure 5.7 shows a histogram plot of the video durations in each of the different categories.

YouTube places a cap of 10 minutes on video length.[1] 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 5.7 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. This spike in videos that are between

3 and 5 minutes in length may be attributed to the large number of music videos that are contained in the list of all time most popular videos. Compared to videos in the all time category, we find longer duration videos in the daily and weekly popular categories. Table 5.5 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.* [42]. Their study had found that the median size of video clips on the Web was about 2 minutes.

5.2.3 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 5.8.

From Figure 5.8 several observations can be made. We find that, among the videos accessed, few 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.

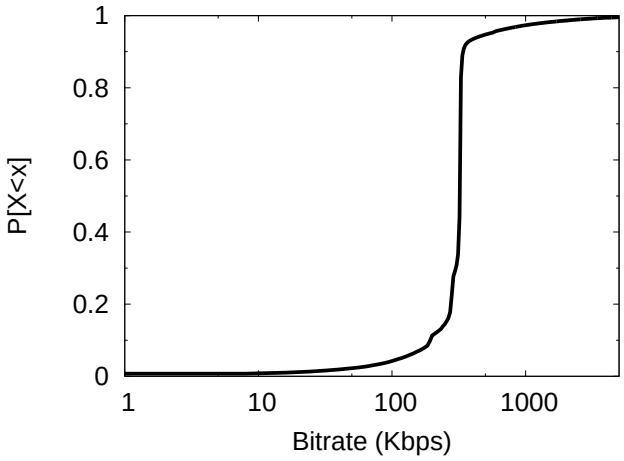


Figure 5.8: CDF of Video Bit Rates (Campus)

Approximately 97% of the videos seen on campus have bit rates below 1 Mbps. Many 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.* [42] 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.

5.2.4 Age of YouTube Content

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. Information about the age of content consumed by Web 2.0 users can be exploited when developing strategies to handle the

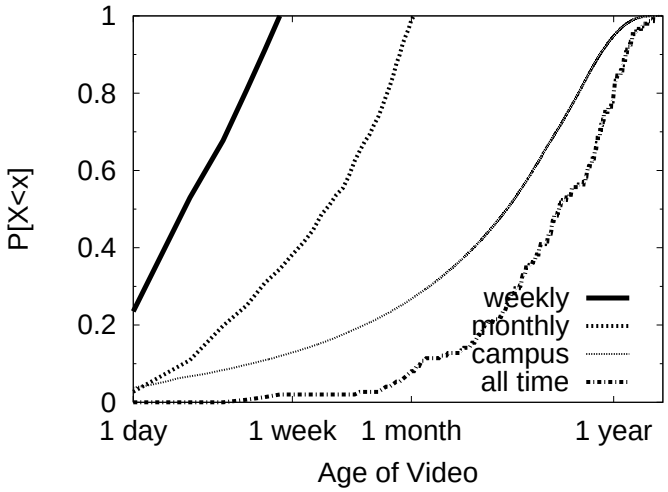


Figure 5.9: CDF of Age of Video Content

resource demands of YouTube users. Specifically, caching and prefetching policies may be developed based on the age of content preferred by YouTube users.

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 5.9 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. This finding has implications for prefetching policies. For example, if an edge network were to deploy a CDN node, prefetching new videos as they are posted

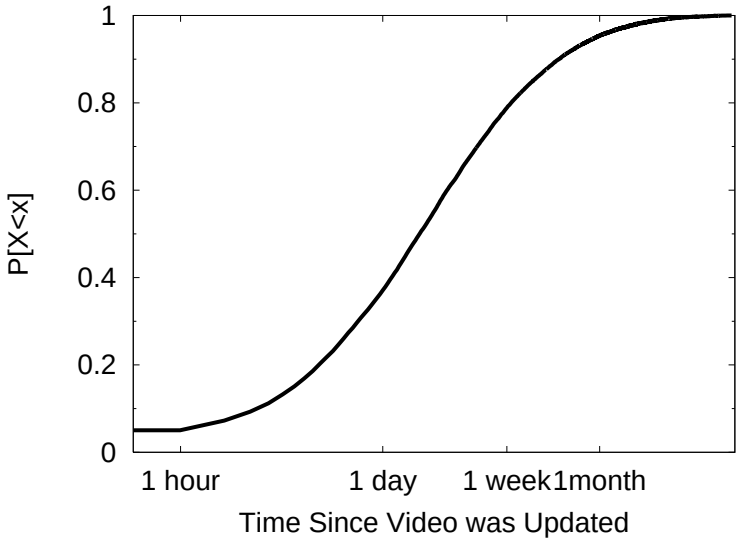


Figure 5.10: CDF of Time Since Video Update (Campus)

would not yield the best performance.

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 5.10 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.

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 5.11 we consider the time since modification for various

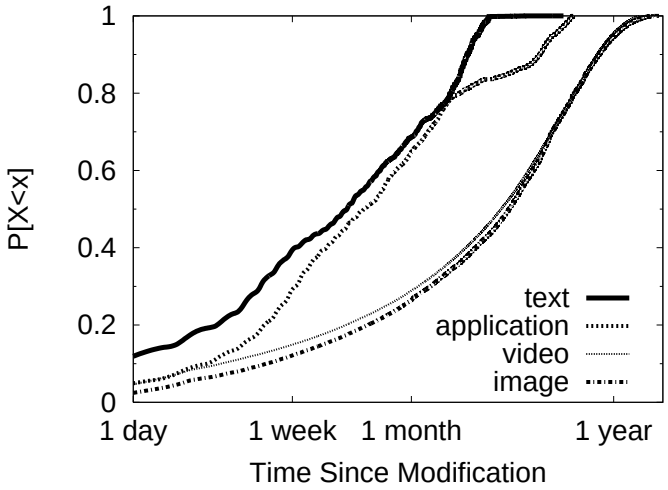


Figure 5.11: CDF of Time Since File Modification (Campus)

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.

5.2.5 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 indicates 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 5.12 where we present a histogram of ratings for unique videos.

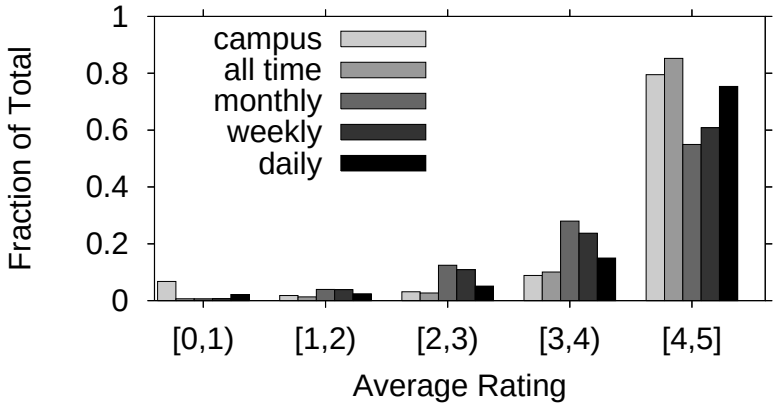


Figure 5.12: Histogram of Average Video Ratings

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 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 help users find content of interest among the large volume of content available at YouTube.

5.2.6 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 5.6. 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 5.6 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

Table 5.6: 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

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. Videos in these categories are enjoyable regardless of their recency. As a result, they are more likely to accumulate the large number of views required to gain a position on the long term most viewed lists. 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.

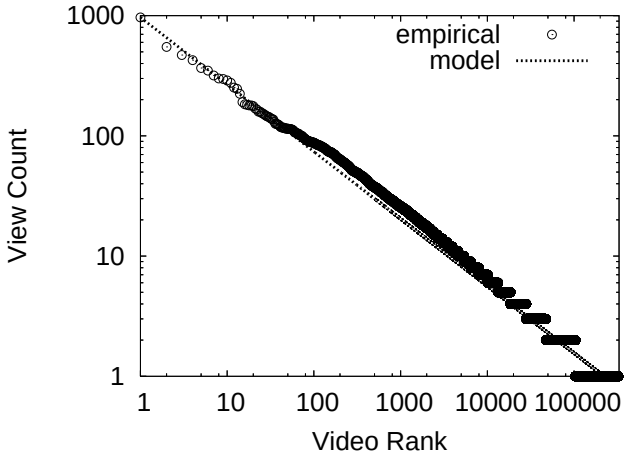


Figure 5.13: Ranked View Count of Videos (Campus)

5.3 File Referencing

The distribution of requests across video content observed on campus is considered in this section. How requests are distributed across content (often referred to as concentration) is useful to consider when determining if traffic management techniques such as caching can be applied to a workload.

5.3.1 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 [72]. Zipf's law has previously been used to model Web document references [7, 11, 46] and media file references [17, 18, 62, 69].

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 5.13 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.* [11] and Mahanti *et al.* [46] 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.

5.3.2 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 behavior was applied previously to understand memory and file referencing behavior [13,47], Web document referencing behavior [7, 46], and more recently to the referencing behavior of media files on an on-demand streaming system [69].

Figure 5.14 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

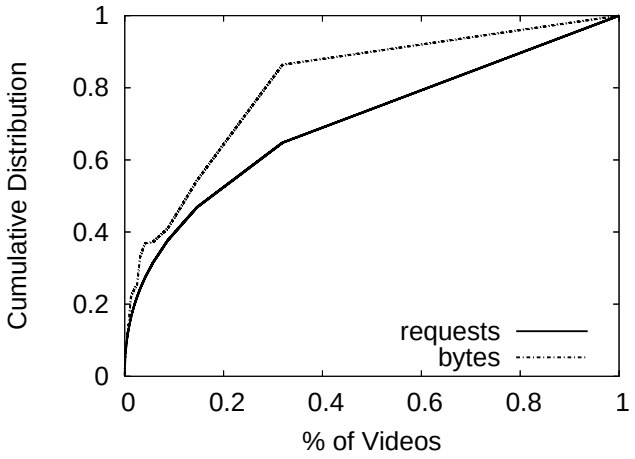


Figure 5.14: Concentration of Video References (Campus)

studies [7,46,69], 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 [11, 46].

We also analyzed occurrence of one-timer videos, that is videos that are requested only once in the entire data collection period. We found 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 documents, 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, 46].

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 specialized and have more limited audi-

ences (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 [46].

5.4 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 [23]. Similarly, locality has also been found to occur in Web server and proxy document reference streams [7, 45, 46]. 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.

5.4.1 Working Set Analysis

Working set can be defined as the set of information accessed by a process in a particular timeframe [23]. In this thesis, working set refers to the set of videos accessed by YouTube users in our edge network dataset. 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 5.15. We observe that the number of requests in common with the first weeks is sensitive to the lower

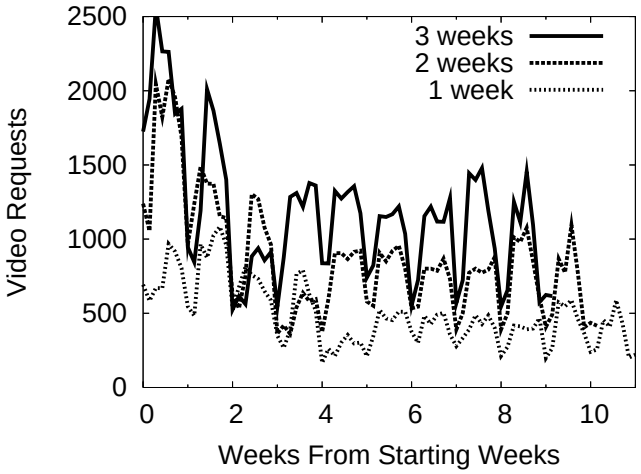


Figure 5.15: Absolute Drift From First Weeks

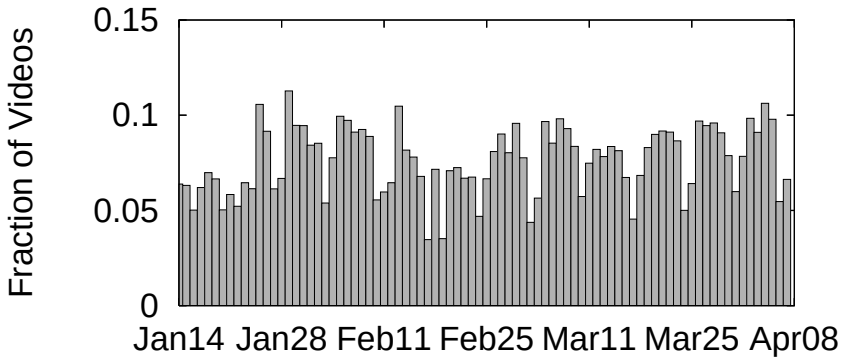


Figure 5.16: Fraction of Previous Days Video Requests Observed

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 5.16 considers short term temporal locality in the set of videos viewed each day (working set). This thesis uses commonality between consecutive days

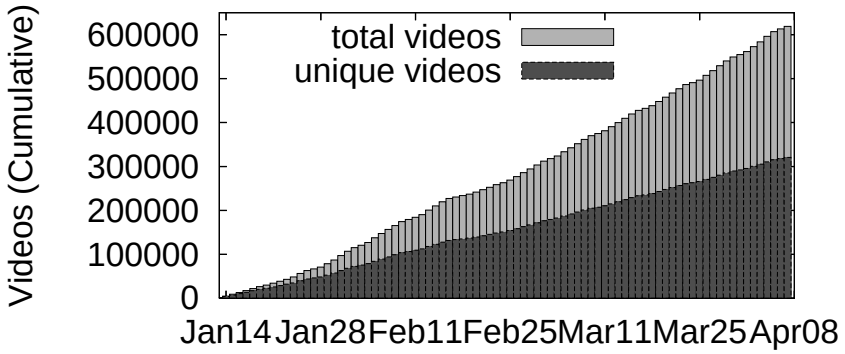


Figure 5.17: Unique and Total Video Growth

as an intuitive metric for short term locality. An alternate metric for short term temporal locality is the Least Recently Used Stack Model (LRUSM) [45]. We find that there is not a very strong correlation between videos viewed on consecutive days. In general, 10% of the previous day's 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, commonality of requests on consecutive days is usually close to 5%. However, as interest in YouTube increased in early February we noticed commonality increase 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 5.17. 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

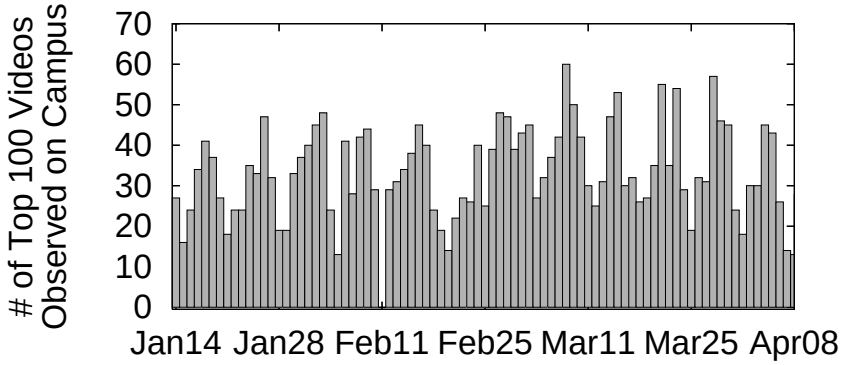


Figure 5.18: Overlap Between Globally Popular and Campus Videos

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 5.1, this would translate into a savings of 3.19 TB.

5.4.2 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 5.18. 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 visiting YouTube. It may be the case that users are directed to YouTube by friends sending them links to 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.

5.5 Transfer Properties

We now consider properties of the transfers observed during our measurement period. Transfer properties such as transfer size and duration have implications for network providers that need to plan for the bandwidth demands of data transfers from Web sites like YouTube. Transfer durations also have implications for server administrators who need to take into account transfer durations when provisioning CPU resources at central servers.

5.5.1 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 transactions, 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 5.19 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

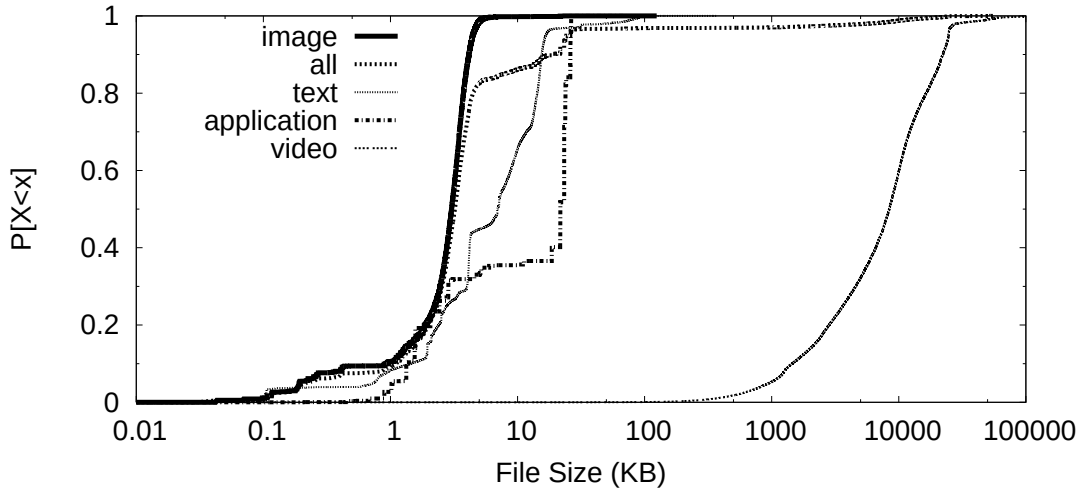


Figure 5.19: CDF of Transfer Sizes (Campus)

on almost every page of the YouTube site. Our results suggest that these images are each typically less than 5 KB 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.

5.5.2 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

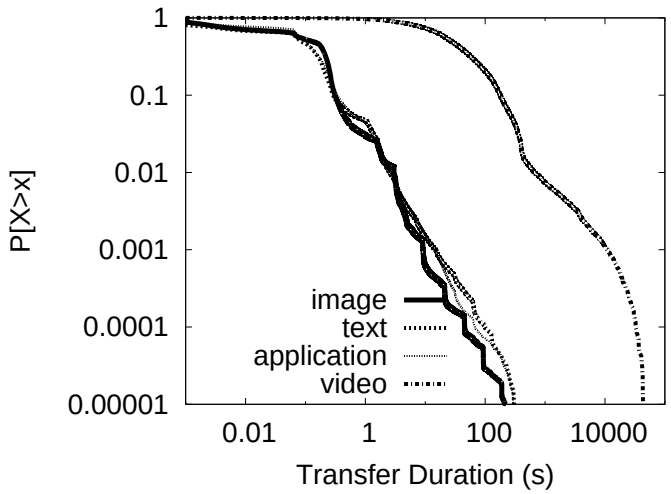


Figure 5.20: CCDF of Transfer Duration (Campus)

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 5.20 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.

5.6 Summary

This chapter considered characteristics of YouTube traffic over our 85 day measurement period where we observe more than 23,000,000 valid HTTP transactions. We

find that while video content only accounts for a small number of transactions, it makes up 99% of the data transferred during our trace. This can be attributed to the large file and transfer sizes of video content when compared to the other content types. As expected, we observe diurnal trends in our trace data with most of the data being transferred during the day on week days.

When considering the popularity characteristics of video content, we find that the popularity of video content consumed by YouTube users follows a Zipf-like distribution. However, the concentration of references we observe is low, suggesting that in order to cache video content locally, a larger cache would be required to achieve the same level of effectiveness when compared to traditional Web content. The large size of video content compounds this issue.

Chapter 6

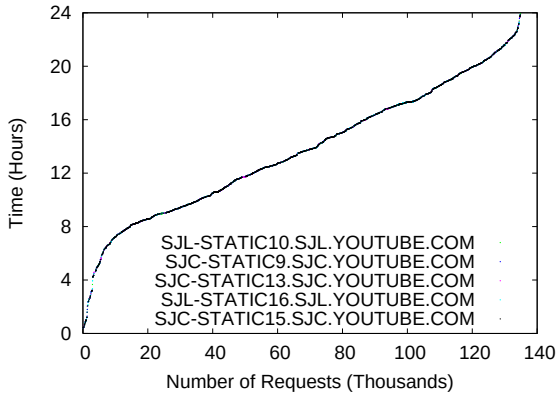
Characterization of Campus User Behavior

This chapter considers characteristics of campus YouTube usage at the user and session level. Challenges faced when using our data sets to analyze user behaviors, as well as our solutions to these issues are presented in Section 6.1. A user level analysis is presented in Section 6.2. The concept of a user session is defined, and user sessions identified within our traces are characterized in Section 6.3. The contents of this chapter are summarized in Section 6.4.

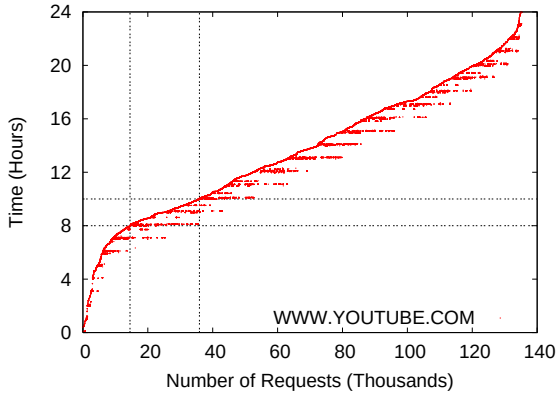
6.1 Methodology Challenges

The goals and challenges of our measurement methodology were discussed in Section 4.1. Decisions made when determining how to deal with the challenges of measuring campus YouTube traffic had impacts on our ability to analyze user behavior. This section discusses the two main issues we encountered. The first issue involving mapping of the visitor ID was a foreseeable limitation that we were not able to overcome. The second issue involving invalid time stamps from the remote servers was not foreseeable. However, it highlights the importance of maintaining a central time stamp in network measurement. We were able to overcome this problem by using a simple method to adjust the invalid time stamps.

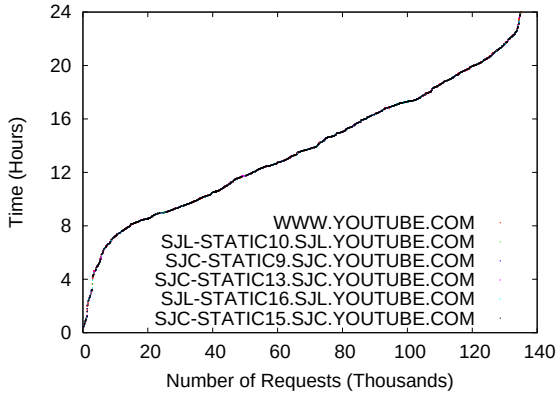
The first issue faced when analyzing campus user behavior comes as a result of the data collection process (`bro`) being restarted every 24 hours to enable log files to be moved from the monitor to an alternate storage space. Recall that within each log file the visitor ID field was mapped to a unique integer to ensure user privacy. To further protect the privacy of campus YouTube users, we opted to reset the mapping



(a)



(b)



(c)

Figure 6.1: Time series of: (a) unaffected servers; (b) affected server before adjustment; (c) affected server after adjustment [31]

with the data collection process rather than recording the mapping to disk. As a result, each visitor ID mapping is only valid for a 24 hour period.

Another aspect of our data collection that posed a challenge to analyzing user behavior was the time stamp we recorded. When we determined the data to record in each transaction, we wanted to utilize a time stamp recorded by `bro` to indicate when the transaction occurred. Unfortunately, at the time of collection there was no network time (`NTP`) service available on the isolated network where our monitor is located. Since we knew our monitor's clock tended to drift, we instead decided to utilize the `Date:` header from the HTTP response as the time stamp for the transaction. We expected that YouTube would synchronize the clocks on their servers, and that the only drawback would be the coarse-grained (1 second) resolution of the timestamps provided by the `Date:` field.

In our trace we observed over 2,000 distinct servers (according to the `Host:` header values) responding to requests for objects on YouTube's site. Most of these servers appear to have synchronized clocks. For example, Figure 6.1(a) shows the behavior of the clocks of several servers in the January 14th trace (based on the `Date:` values each server provided). For each of the 135,231 transactions recorded on that day, the graph plots a point if the transaction was served by one of the identified servers. In Figure 6.1(a), the timestamps from transactions of five of the busier servers are shown (these servers account for 3.0%, 2.5%, 2.4%, 2.2% and 2.1% of the replies served on January 14th, respectively). As can be seen, the timestamps never decrease in value (there may be several with the same value, if a burst of replies happen at the same time), and appear to be well synchronized across these servers.

However, the busiest server (www.youtube.com, which served 32% of the requests) frequently reported timestamps in the past; Figure 6.1(b) shows the behavior of the timestamps for this server over the course of the trace. This figure shows that the timestamps do not always behave as expected. For example, between re-

quests 14,500 and 35,900 there are many responses with a time stamp of hour 8 (as illustrated by the horizontal line at y=8), even though for other responses the timestamps (by request 35,900) are at hour 10. A related observation is that there only seem to be problematic timestamps in the past. The source of this problem is not entirely clear; if for example, there were multiple servers handling requests for www.youtube.com and their clocks were not synchronized, we would expect to see more variation in the timestamps, both ahead of and behind the correct time. From Figure 6.1(b) it appears that some timestamps are quite common, possibly because the application was reusing a stale cached time stamp.

Our solution to this problem is rather simple: whenever a time stamp was observed in the past for a server, that time stamp is replaced with the last time stamp value that was deemed to be correct (i.e., the "current" time for that server). If a value in the future is observed, then that values becomes the "current" time for that server. This approach updates the time stamp based only on values a server itself reported. Specifically, we do not utilize information from one server to adjust the time values of another server. This prevents one misbehaved server from affecting all other servers in the trace.

For validation purposes, we compare the adjusted timestamps for www.youtube.com to the original timestamps reported by the other servers in the trace. The results are shown in Figure 6.1(c). From this figure, we see that the timestamps (and thus the clocks on these servers) appear to match, even though we did not utilize the information from these other servers to update www.youtube.com. Thus, we conclude that our method for adjusting the timestamps was reasonably accurate, and does not bias our results in any significant manner.

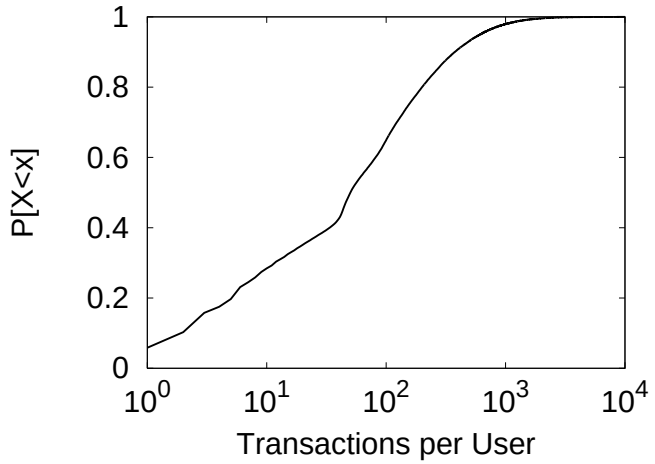


Figure 6.2: CDF of Transactions per User

6.2 User Level Characterization

In this section we characterize the behavior of campus YouTube users (as identified by the visitor ID field) during our trace period. We consider the resource demands of users by analyzing the amount of data they transfer and the number of transactions they complete. These simple user level characteristics can provide useful insights for capacity planning. By combining knowledge of user population growth (cf. Figure 5.1) with knowledge of the workload generated by each user, it is possible to predict when a link may be saturated and should thus be upgraded.

We first examine the amount of network resources consumed by campus YouTube users. The cumulative density function (CDF) of transactions generated by each user is shown in Figure 6.2. The mean and median number of transactions per user are 152 and 51, respectively. We find that the transactions generated by each user are highly variable, with a coefficient of variation equal to 4.5. This is a result of some users issuing very few requests while others issue thousands of requests. We observe that the users with few transactions are likely not interested in the content of YouTube, as 75% of users who have less than 100 transactions transfer 1 or no videos. We

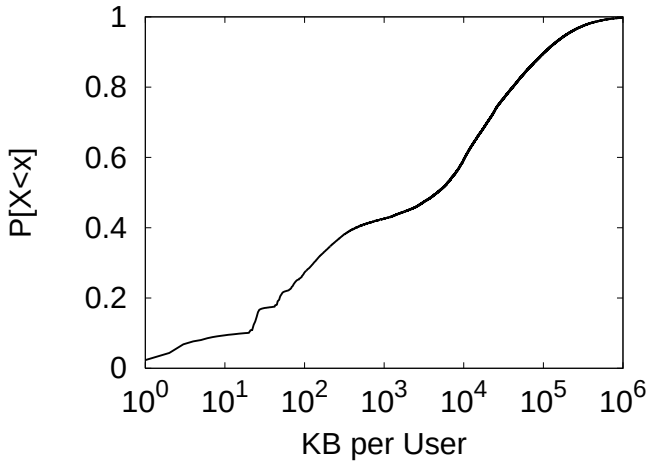


Figure 6.3: CDF of Bytes per User

find that many of these users are referred to YouTube by third party Web sites that embed YouTube videos (using `html` tags). This embedding results in sessions with 1 or fewer videos viewed. We observe the effects of this embedding on user sessions in Section 6.3. In contrast, the users who issue more than 100 transactions seem more engaged by the video content, with 50% of them transferring more than 5 videos.

The amount of data transferred by each user is considered in Figure 6.3. We find that the mean and median amount of data transferred are 40,988 KB and 4,695 KB, respectively. Similar to the number of transactions per user, the amount of data also displays high variability with a coefficient of variation of 3.1. This variability can be attributed to the large size of videos and the variability in user viewing habits. For example, 75% of users who view at least one video transfer more than 8 MB of data. In contrast, the median number of bytes transferred by users that do not watch any videos is 51 KB.

6.3 Session Level Characterization

In this section, we consider user behavior at the session level. In Section 6.3.1 we define the notion of a session and our methodology for analyzing them. Session durations are considered in Section 6.3.2. Time of day effects on concurrent sessions and inter-session times are presented in Section 6.3.3. Finally, inter-transaction times and content types transferred by user sessions are discussed in Sections 6.3.4 and 6.3.5.

6.3.1 Defining YouTube User Sessions

A user session is defined by Menasce *et al.* as a series of requests issued by a user to a Web site in a single visit to the site [50]. In the case of YouTube, a user session may include browsing lists of videos, searching for specific videos, viewing selected videos, and interacting with others through comments and ratings. These actions differ from sessions to conventional Web sites, which usually do not feature multimedia and interaction to the same degree as Web 2.0 sites such as YouTube.

Determining the start and end of a user session by observing network traffic can be challenging, particularly for popular sites such as YouTube, where repeat visits are common. Since there are no distinctive login and logout transactions in most sessions, we instead utilize a timeout threshold to determine when a session for a specific user has ended. Any subsequent requests from the same user are then considered to be part of a separate (new) session. In other words, two consecutive transactions are considered to be a part of the same session if the time between them is less than the threshold value. If the time between two transactions is greater than the threshold value, the latter transaction would be considered to be the first transaction of the next session.

Choosing an appropriate value for the threshold is important when analyzing

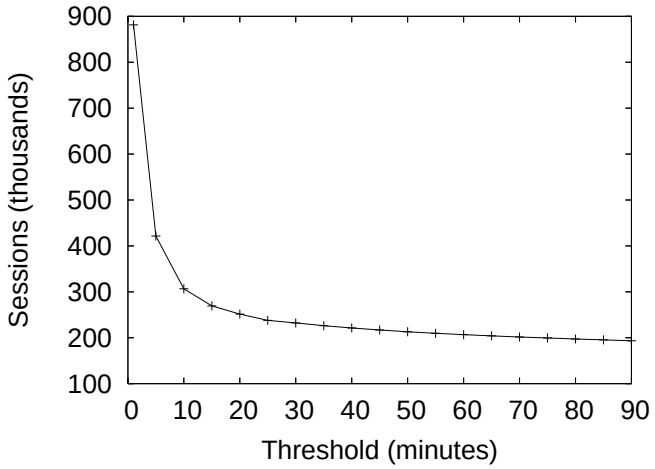


Figure 6.4: Total Sessions Observed for Various Timeout Thresholds

user sessions in this fashion. If the threshold is too large, multiple distinct sessions may be incorrectly combined into a single session. Conversely, if the threshold is too small, each actual session could be fragmented into a number of small sessions (in the extreme case, many sessions each containing a single transaction).

We evaluate multiple timeout thresholds to determine which one is appropriate for our workload. Figure 6.4 presents the total number of sessions for a variety of threshold values. An extreme threshold value that is too small can be observed for the threshold value of one minute; using this threshold results in 881,324 sessions for our dataset. As the threshold increases, the number of sessions observed continuously decreases. However, once a threshold value of 40 minutes is reached the total number of sessions begins to level off. This suggests a value of 40 minutes may be a reasonable threshold. To test this hypothesis, we plot the number of sessions observed per user for various threshold values in Figure 6.5. As the threshold is increased, we observe an increase in the number of users having only one session. This is to be expected since each user's identity is only valid for a 24 hour period. The difference between the distributions for the different threshold values is large for small threshold values,

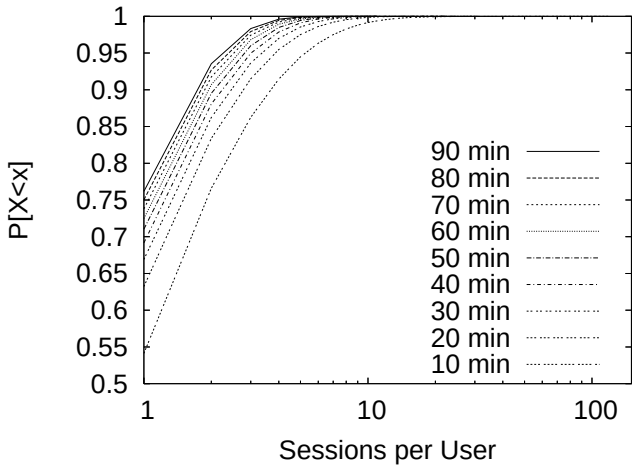


Figure 6.5: CDF of Sessions per User Observed for Various Timeout Thresholds

but it becomes more consistent around a threshold value of 40 minutes.

Based on this analysis of threshold values, we conclude that for our traces a timeout value of 40 minutes is appropriate for delimiting sessions. When compared to previous work on user sessions for conventional Web sites (e.g., [5, 57]) the threshold value we observe is much larger (e.g., our timeout is 4 times larger than the 10 minute timeout value used in [57]). Our large threshold value may be attributed to multiple factors. These include the time it takes a user to view a video and the plethora of content that keeps users at the site longer than is the case for traditional Web sites. The remainder of the analysis in this paper uses a threshold of 40 minutes to distinguish between multiple sessions from individual users.

6.3.2 Session Duration

The duration of user sessions serves as an indication of the level of engagement users have with a Web site. This is of interest to advertisers, but also to network and server capacity planners who must be aware of the resource demand for each user, and any changes that occur over time. Long sessions may indicate that the Web site

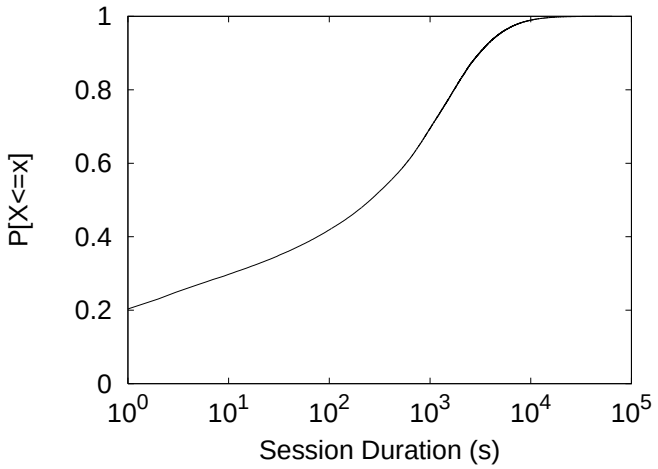


Figure 6.6: CDF of Session Durations

is providing content that the user finds interesting and will browse and watch for sometimes hours on end. Shorter sessions, however, may indicate low engagement or an unsuccessful visit (e.g., searching for a certain video but not finding the video).

The CDF of session durations is presented in Figure 6.6. User session duration is calculated as the difference between the end time of the last transaction (time the last transaction starts + its duration) and the time the first transaction of a session was observed. Due to our log rotation at 4:30 am, sessions that are active during the rotation may be split into 2 separate sessions. Since the rotation takes place when usage is typically the lowest in a given 24 hour period, the impact of this rotation is expected to be minimal.

We observe that the duration of YouTube user sessions are similar to those observed in previous work [5]. The mean session duration is 18.7 minutes and half of the sessions exceed 4.3 minutes. We observe a coefficient of variation of 2.1 suggesting that user engagement in YouTube is highly variable. While our session durations are similar to previous work, it is important to consider the nature of the Web site studied by Arlitt [5]. In that study, the Web site for a popular sporting event was

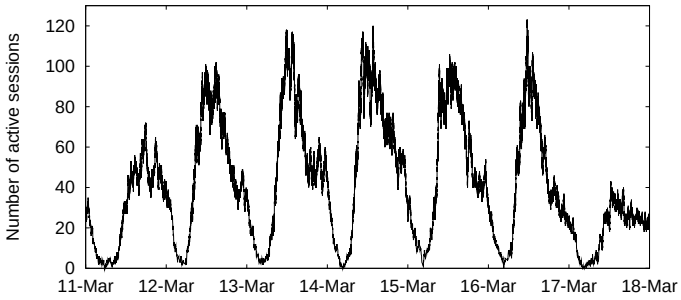


Figure 6.7: Concurrent Sessions Over the Course of a Week

studied. Presumably, this Web site offered live scoring of events which may have increased the time users spent at the Web site. As a consequence, while our results may resemble those of a traditional Web workload, the workload we are comparing to may have been biased towards longer sessions than other traditional Web sites. We also observe some sessions with very short duration in our data. These are caused by other Web 2.0 sites embedding YouTube content. When a user views a page that includes an embedded video, they may not choose to not view this video. However, regardless of whether or not the video is viewed, the flash player and some thumbnail images are transferred by the user's browser, resulting in a short session with no video transfers.

6.3.3 Active Sessions and Inter-session Times

Since the number of users varies over time (cf. Section 6.2) we expect similar influences on the number of active sessions. The number of active sessions over the course of a week is presented in Figure 6.7. We present results for the week of March 11, however, we note that this week is qualitatively similar to other weeks during our measurement period. As expected, we observe a higher number of concurrent sessions on the weekdays (March 12-16) where the daily peak in concurrent sessions is around 120 active sessions. In contrast, on the weekend the daily peak is closer

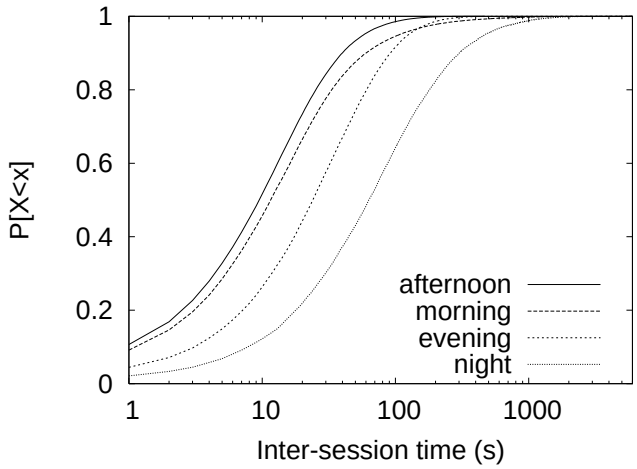


Figure 6.8: CDF of Inter-session Times

to 80 concurrent active sessions. We also observe variability in the number of active user sessions between the various times of day. After midnight the number of sessions steadily decreases in the late night hours. When the workday resumes we observe the number of active sessions increasing throughout the morning and afternoon with decreasing numbers of active sessions in the evening hours as most campus users go home for the night.

As we have seen, the daily routine of campus users impacts the number of active sessions. We now consider inter-session times across various times of day. The times of day we consider are: night (midnight-6 am), morning (6 am-12 pm), afternoon (12 pm-6 pm) and evening (6 pm-12 am). The CDF of inter-session times across the various times of day is presented in Figure 6.8. As expected, we observe the shortest times between session arrivals in the afternoon, when the mean and median inter-session transactions are 17 and 10 seconds, respectively. The morning and evening periods show transitions between the busy afternoon period and the much less busy night period. The mean inter-session times for morning and evening are 32 and 39 seconds, respectively. Finally, night has the longest inter-sessions time, with a mean

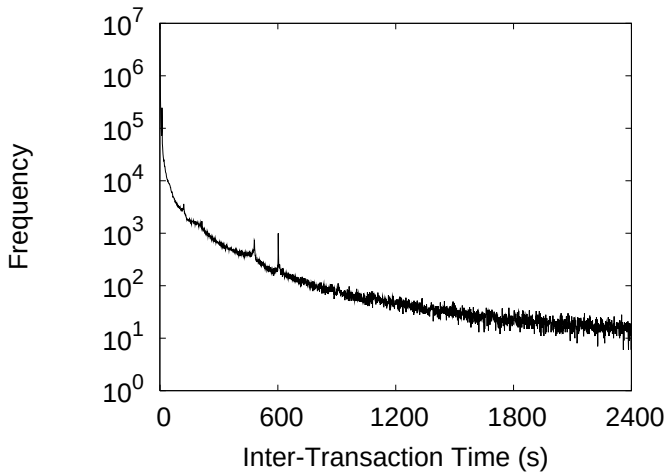


Figure 6.9: Frequency Distribution of Inter-transaction Times

inter-session time of 130 seconds.

6.3.4 Inter-transaction Times

Since sessions are composed of a set of transactions it is interesting to consider the time between transactions, referred to as the inter-transaction time, within a session. Inter-transaction time can provide insights into the behavior of users as they browse the YouTube site, as well as the automated requests generated by the user's browser. The frequency distribution of inter-transaction times is presented in Figure 6.9. As a Web 2.0 site that uses `AJAX`, YouTube requires that several files be loaded to display a Web page. These automated requests can be observed in the large number of very small inter-transaction times. Specifically, only 14% of inter-transaction times exceed 1 second. The larger inter-transaction times are more likely a result of user think times as they view a page and then consider which page to visit next, thus generating a new transaction. We observe that user think times are larger for YouTube than has been observed in previous studies of traditional Web workloads [5]. This may be attributed to the time it takes users to view a video on a specific page and then

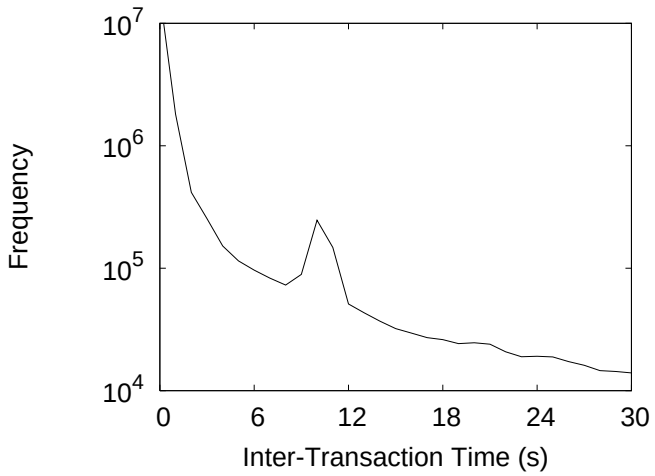


Figure 6.10: Frequency Distribution of "active" OFF Times

move on to browse more pages.

Automated behavior caused by the user's browser can be modeled using an ON/OFF model as is done in previous work [8]. In this model, ON periods describe the time spent transferring objects and the OFF periods are the time between object requests. Two types of OFF periods can be used to distinguish user and machine generated requests; "active" OFF periods denote the time between transfer of objects embedded within a single Web page (i.e., the requests are being generated by the user's browser in order to render the Web page) and "inactive" OFF periods that characterize the time between use generated requests for Web pages. When generating a synthetic Web workload it is important to take both of these types of OFF periods into consideration to capture behavior of both the user and their browser.

We consider the "active" and "inactive" OFF model in our trace data. To separate the "active" and "inactive" OFF times we use a threshold value. Based on the frequency distribution shown in Figure 6.10 we select a threshold value of 30 seconds. Since we collect timestamps from the `Date` field in the HTTP header our

timing granularity is 1 second. While 30 seconds is a larger threshold than was previously considered [8], we find that it is appropriate for our traces. The larger threshold value is able to encompass automated aspects of Web sites that typify Web 2.0. Specifically, `AJAX` can increase the time between automated requests as Javascript files must be transferred and processed before any embedded requests within the Javascript can be made. There is also the functionality to control timing of requests using Javascript. For example, after a video is played, thumbnail images for similar videos are displayed to the user, rotating two images at a time. These images are transferred approximately every 10 seconds and are likely the cause of the large number of inter-transaction times between 9 and 12 seconds. Following this aforementioned automated activity that we observe in the 9-12 second range, the distribution of inter-transaction times begins to level off at 30 seconds, leading us to select 30 as the threshold for our "active" OFF times.

6.3.5 Content Types

We now examine the types of content transferred during YouTube sessions. Content types are determined using the `Content-Type:` field in the `HTTP` response. The content types we consider in our study are the following:

- applications (e.g., application/javascript, application/xml, application/x-shockwave-flash)
- images (e.g., image/jpeg, image/png, image/gif)
- text (e.g., text/html, text/css, text/xml)
- videos (video/flv)

The CDF of the transactions per session for each content type is shown in Figure 6.11. We observe that images are the most prevalent content type transferred,

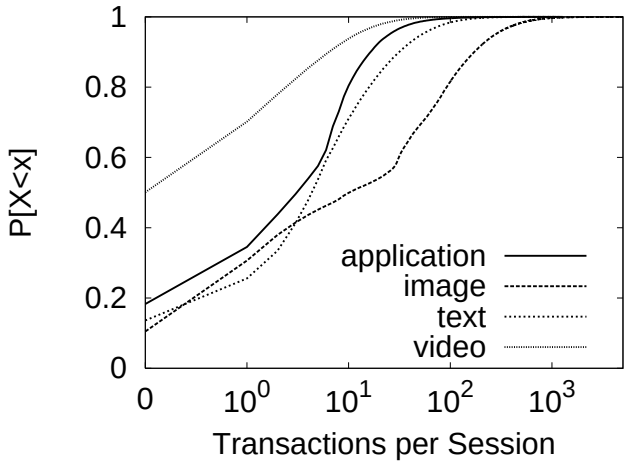


Figure 6.11: CDF of Transactions per Session

with an average of 67 images transferred in each session. This is not surprising as many users will browse the various pages of videos which results in the transfer of a thumbnail for each video listed on a given page. The next most prevalent content types are text and application with averages of 13 and 8 transfers, respectively. These correspond to, for example, individual pages a user views (e.g., a video catalog page), and a video detail page which loads the video player plugin. Interestingly, video content has the lowest number of transactions within each session, with an average of just three videos transferred within each session. The average number of videos transferred is lower than one might expect due to the large number of sessions that do not transfer any videos. We find that 51% of sessions do not transfer any videos. Given that YouTube is a site that centers around video content, the large number of sessions that do not transfer any videos seemed unusual at first. Upon further inspection, we find that over half of the sessions that transfer no videos were referred to YouTube from other Web sites with embedded YouTube content. We also notice a large number of sessions that do not transfer any application, image or text data. This is likely a result of these content types being cached.

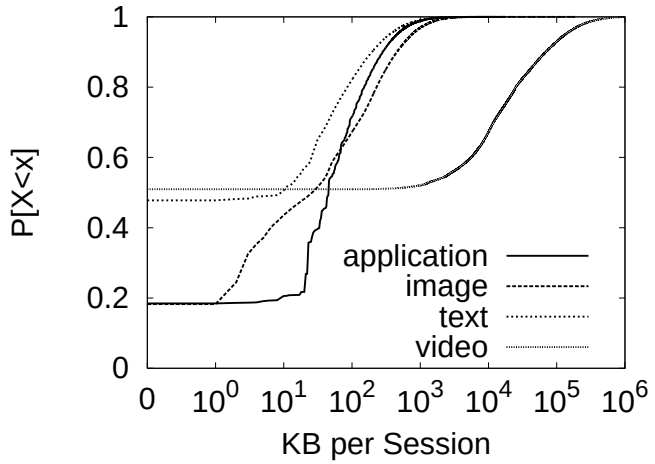


Figure 6.12: CDF of Bytes per Session

We now turn our attention to the bytes transferred for each content type within a session. The CDF of bytes transferred for each content type is presented in Figure 6.12. Despite having the fewest transactions within a session, video transfers dominate the bytes transferred during sessions, with a mean of 27 MB. Image and application transfers follow video for having the most bytes transferred within sessions. The average number of bytes transferred for image and application are 169 KB and 115 KB, respectively. Given the large number of thumbnails and Javascript that must be transmitted to display a YouTube video page, it is not surprising that these content types make up a large portion of the transactions and bytes within user sessions. In general, text constitutes the fewest bytes transferred within a YouTube session, with an average of 78 KB.

6.4 Summary

In this chapter, the behavior of campus YouTube users was presented. Issues faced when using our data sets to analyze user behavior were highlighted. They included a lack of long-term user data as a result of measures taken to increase user privacy as

well as problems resulting from relying on remote servers to provide accurate time stamps. While both of these issues were undesirable, they nonetheless provided us with valuable insights for future measurements of Internet traffic.

Taking into consideration these measurement issues, a characterization of campus YouTube users, and user sessions was undertaken. Several characteristics of user sessions were considered including, duration, inter-session times, and content types transferred. An interesting phenomenon that we observe in our traces is a large number of campus YouTube users do not directly visit YouTube, but are referred to it by third party Web sites that embed YouTube content using `html` tags.

Chapter 7

Discussion

In recent years, developments in Web technology have facilitated a change in how the Web is used. These developments combined with the widespread availability of residential broadband connections have enabled a new generation of Web sites to proliferate and offer services that were previously not available on the Web. User generated content, social networking and software provided as an online service are among the services provided by this next generation of Web sites. In this thesis, we characterized one of the most popular Web sites providing user generated content, YouTube.

The observations made in this thesis provide insights into the characteristics of Web traffic associated with user generated content. These observations have implications for many different parties. First, with an understanding of the resource demands of a site like YouTube, network providers are better able to provision resources and develop strategies for managing network traffic. Second, content providers that hope to develop sites that provide user generated content can allocate appropriate hardware based on our observations of the storage and processing requirements of YouTube traffic. Finally, our observations can be applied to developing models of the Web traffic associated with Web sites that center around user generated content. Implications for each of these parties are discussed in Section 7.1.

While the work in this thesis (and the resulting publications [30, 31]) is one of the first characterization studies of YouTube, there have been some parallel studies that focus on the global characteristics of YouTube usage by using Web crawling techniques [15, 16, 36]. More recently there has also been another study of YouTube

usage from an edge network perspective [71]. Section 7.2 compares our results with those from these parallel studies.

7.1 Implications

In this section we describe the significance of the results presented in Chapters 5 and 6. In Section 7.1.1 we discuss issues for (edge) network providers. In Section 7.1.2 we examine implications for service providers. Finally, the implications of our findings for modeling Web traffic are discussed in Section 7.1.3.

7.1.1 Implications for Network Providers

Web 2.0 sites that provide user generated content have proven to be extremely popular. In particular, there is a demand for multimedia user generated content such as videos. The popularity of multimedia user generated content places significant demands on the bandwidth of edge networks as this multimedia content is much larger than other content types found on the Web. Caching and CDNs have previously been used to handle the network resource demands of Web traffic. This section examines the applicability of each of these approaches for network providers.

Web caching is a technique for managing Web traffic that has many benefits. By moving content to caches that are closer to the user than the original content server, Web caching reduces bandwidth consumption of Web traffic, as well as demand on the original content server. The proximity of caches to clients can also be leveraged to improve end user experience through reduced latency when requesting static Web objects. Since Web 2.0 sites such as YouTube use the same application layer protocol as traditional Web content (HTTP), it is possible for previous caching infrastructures to be directly applied to Web 2.0 sites. However, the results of this thesis suggest that using the same logical infrastructure for both Web 2.0 and traditional Web

traffic may not be the best approach.

There are three main factors that motivate our recommendation that Web 2.0 and traditional Web traffic have logically separate caching infrastructures. First, the popularity of multimedia user generated content means that Web 2.0 traffic is likely to include more large multimedia files than traditional Web traffic. As observed in this thesis, while making up only a small portion of requests, video content makes up almost all the bytes transferred for Web 2.0 sites. If a caching infrastructure is to reduce the load on a Web 2.0 server it will be necessary to cache this multimedia content. If the same caching infrastructure is used for both Web 2.0 and traditional Web traffic these large multimedia files may displace many smaller objects. This could, for example, decrease cache hit rates and potentially degrade the experience for many other users. Second, Web 2.0 sites facilitate posting of user generated content by a wide range of users. This results in a repository of diverse multimedia content that is able to serve a population of users with diverse interests. This diversity in content and interest necessitates that more objects be cached for Web 2.0 sites in order to achieve a reasonable byte hit rate. Third, object replacement algorithms for caching Web 2.0 may differ; while some characteristics (e.g., recency, frequency) may still be important, others 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.

CDNs are another technique for managing Web traffic that emerged in the late 1990's. CDNs provide many of the benefits of Web caching, and give content providers more control over their content. In particular, by locating CDN nodes closer to their users CDNs are able to improve end user experience by reducing latency. In the case of clients located far away from the original content server this can be a significant improvement. CDNs are an alternative to caching to reduce

the impacts of Web 2.0 on 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. The breadth of multimedia user generated content makes finding an appropriate method for populating a local CDN node challenging. Also, our observation that the most popular videos on YouTube do not make up a significant amount of campus YouTube traffic suggests that a prefetching method that takes into account local popularity of content would be more successful than one that focuses on global popularity.

7.1.2 Implications for Service Providers

A key component of many Web 2.0 sites is user generated content. The notion of user generated content decouples content creation from content hosting (which is done by a service provider such as YouTube). Content creation is highly decentralized with users from around the world creating and posting content to Web 2.0 sites. This decentralized content creation 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, posting multimedia content was often expensive with the content provider having to pay for storage to store their content in addition to bandwidth required to serve the content to users. As a result of this, interesting online multimedia content was often limited. In recent years, there has been an increase in online multimedia as business models (e.g., iTunes) have been

developed. Web 2.0 has facilitated even more widespread proliferation of online multimedia content. A key component of many Web 2.0 sites is social networking. Social networks enable communities to grow around services such as video sharing. Given the relative ease of digital content creation (e.g., text, photos, videos) and sharing this content (using services like YouTube) many more users are able to create and share online content. With this ease of creation and sharing, 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 [43]; with an average size of 10 MB for each video (Table 5.4), this means YouTube's video repository grows by approximately 19.5 TB per month! Furthermore, if the user base increases, more 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) will be preferable for hosting Web 2.0 sites.

Workloads for sites such as YouTube also have implications for the choice of server used to store the content. For example, as we have observed in Section 5.5.2, serving larger objects such as videos occupies the server for a longer period of time and utilizes more CPU cycles. Since these longer duration transactions occupy an HTTP server process or thread for longer periods of time, the concurrency of the server will be an important factor to consider. Such workloads are better served using multi-core systems than traditional single-core systems. Multi-core systems are better able to support many 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.

7.1.3 Implications for Modeling Web Traffic

This thesis characterizes the workload created by users of a popular, multimedia based, Web 2.0 site. As Web 2.0 sites grow in popularity, it is important to consider the similarities and differences between Web 2.0 and traditional Web traffic in order to understand the current properties of Web traffic. The different workload demands of Web 2.0 sites are important to consider for development of synthetic workload generators. The continuously changing nature of Web workloads and the development of Web 2.0 demand that these changes be considered if a workload generator is to be representative of current Web traffic.

A technological driving force behind Web 2.0 is `AJAX`, which combines Javascript and XML with traditional HTML to provide a rich user experience. We observe the effects of this technology in our observations of inter-transaction times within user sessions. The majority of transactions in our study are generated automatically by the user's browser. Also, the time between automated transactions is larger than in previous work [8]. This is as a result of more complex Javascript issuing requests at specifically timed intervals. As technologies like `AJAX` gain popularity, the automated behavior of these scripts may have a greater effect on the timing of Web requests. As a consequence, it is imperative to develop user models that encompass both automated and human behavior.

User generated content is a key component of Web 2.0 that determines the types of files and amount of data transmitted by users. While images are the most commonly transferred file type, the large size of videos causes them to contribute the most to the number of bytes transferred by user sessions. Video transfers result in the average data transfer of users being much larger than for conventional Web sites, thus placing additional stress on network resources. Modeling the larger file sizes and types of files transferred by Web 2.0 users is important when developing workload generators

to evaluate new platforms and techniques for handling Web traffic.

7.2 Comparison with Parallel Work

This section compares our observations of file properties, file referencing, locality characteristics, and transfer properties to observations made in parallel studies of YouTube. Studies we compare with take two main approaches to analyzing YouTube content. While the majority consider global characteristics of YouTube obtained using Web crawling techniques [15,16,36], there has also been a study that considers YouTube characteristics from an edge network point of view [71]. This thesis employs an approach that combines edge network measurement with study of the properties of videos that are globally popular on YouTube.

7.2.1 File Properties

Cheng *et al.* examine properties of videos stored on YouTube using Web crawling to sample YouTube's video repository [16]. The authors of this study consider many of the file properties that are considered in this thesis. These properties include the categories that videos belong to. The authors note that the three most prevalent categories they observe are music, entertainment, and comedy. We make similar observations from our edge network point of view, finding that these three categories are also the most prevalent on campus.

Video bit rate and file size are also studied by Cheng *et al.* They observe bit rates and file sizes that are similar to those observed in this thesis. Specifically, the authors find that a large number of videos they sample have bit rates around 330 Kbps. We find that the mean and median bit rates of videos viewed on campus are 394 Kbps and 328 Kbps, respectively. File sizes of videos observed by Cheng *et al.* have a mean value of 8.4 MB. This is fairly close to the mean file size of videos

viewed on our campus which is 10.1 MB.

It is interesting to note that while Cheng *et al.* take a more central approach by crawling YouTube's video repository, we are able to make similar conclusions by observing YouTube usage from an edge network perspective. This suggests that considering file properties of YouTube traffic from an edge network perspective may be adequate to gain insights into file properties at the server.

7.2.2 File Referencing

File referencing behavior was considered in many of the parallel studies [15, 16, 71]. We find that our observations are consistent with the results of Zink *et al.* [71] who observe YouTube usage at a university campus. They observe 77% of videos accessed on their university campus were accessed only once (one-timers). Similarly we observe that 68% of the videos we observe are requested only once. The slightly lower number of one-timers that we observe in our data sets is likely a result of our long term measurement and larger user population due to our measurements taking place over the course of a university semester. This is in contrast to the measurements made by Zink *et al.*, who took measurements for shorter durations and outside of the regular academic term.

While our insights into file referencing behavior are similar to those made at an edge network, they differ from global observations of file referencing made by studies that sampled YouTube's repository using Web crawling [15, 16]. This contrast in file referencing behavior between the local and global perspectives is not surprising and has been observed in prior studies of Web proxy workloads [46]. The contrast occurs because of the smaller population of users at the edge network when compared with the total population of users that access the server. The smaller, edge network population still has access to the large volume of content available at the server, thus causing the popularity of objects to be more diluted from a vantage point that is

closer to the network edge.

Cheng *et al.* observe Zipf-like behavior when they study the popularity of YouTube videos [16]. They observe a distribution where the head of the distribution follows a Zipf-like distribution with $\beta = 0.54$ but the tail of the distribution drops off sharply due to a very few unpopular videos observed in their study. They reason that the popularity of videos that they observe does not follow a Zipf distribution because of the unpopular content and reason that it may instead follow the Gamma or Weibull distributions. While this is a valid conclusion, it is interesting to note that the sample of videos taken in this study was made by starting with the list of most popular videos, and traversing videos related to the most popular videos. It is reasonable to suggest that the small number of unpopular videos they observe may have come about as a result of their use of the most popular videos list as a starting point for their sampling. Also, the low value of β observed in the popularity distribution of the most popular videos suggests that they observe more very popular videos than would be expected if the popularity followed a Zipf distribution.

Cha *et al.* consider the popularity of user generated content in detail [15]. They observe a large amount of skew in the popularity of videos with the most popular 10% of videos that they sampled accounting for 80% of the views. This level of skew has positive implications for using caching to handle the resource demands of YouTube traffic. When they consider the popularity distribution of YouTube content they observe Zipf-like behavior in the body of the distribution with flattening at the head of the distribution and a steep drop off in the tail of the distribution. The authors of this study seek to find the causes of the non-Zipf behavior at the head and tail of the popularity distribution that they observe. They suggest a "fetch at most once" effect for the most popular videos that has previously been proposed for peer-to-peer systems [34]. The "fetch at most once" effect occurs because unlike conventional Web content, files in peer-to-peer systems are not likely to be fetched

multiple times by the same user. While this effect seems to explain the reduced amount of extremely popular content observed by Cha *et al.*, it can be argued that YouTube's infrastructure would reduce the impacts of the "fetch at most once" effect. This is because YouTube does not allow users to download video content. As a result, if a user wants to watch a video multiple times they need to request it again. When considering the limited number of unpopular files, the authors find that the distribution of popularity is best modeled as a Zipf distribution with an exponential cut off. They note that while the Zipf distribution is *scale free* the exponential distribution is *scaled* and as a result it is unlikely that they would occur together as the result of a single mechanism. This leads the authors to conclude that the cause of the small number of unpopular videos likely comes as a result of filtering effects such as recommendation systems that favor more popular content.

While popularity distribution is important to consider when determining if methods such as caching or CDNs should be used to handle the resource demands of YouTube traffic, locality characteristics are important to consider when determining how to implement these methods. The following section compares our observations of locality characteristics with parallel studies.

7.2.3 Locality Characteristics

This thesis not only uses measurements from an edge network perspective. We also maintain information on the most popular videos on YouTube during our data collection period. When comparing the videos viewed on campus to videos in the most popular lists we find that while approximately half of the most popular videos are viewed on campus, they tend to make up a very small amount of the video traffic observed on campus. Similar observations are made by Zink *et al.* who find that a video's local popularity is not highly correlated with the video's global popularity. This supports our conclusion that making caching decisions based on which videos

are popular on YouTube is not a good strategy when considering caching YouTube content at an edge network.

7.2.4 Transfer Properties

Similar to this thesis Zink *et al.* use edge based measurements from a university campus to study transfer characteristics of YouTube traffic [71]. In general, their conclusions are similar to our own. They find that the average transfer duration for video content ranges between 81-99 seconds in their traces. We observe similar transfer durations with the mean transfer duration for video content observed in our measurements being 104 seconds. The average transfer size Zink *et al.* observe for video content is approximately 7 MB (across their traces); this contrasts with our average transfer size of 10.3 MB for video content. While Zink *et al.* observe slightly smaller video transfers, they are still significantly larger than transfer sizes observed for traditional Web content.

7.3 Summary

This chapter summarized the implications of the results presented in Chapters 5 and 6 for network and content providers, as well as for Web traffic modeling. Key implications include the challenges associated with the large file size of YouTube content and the wide array of content available on Web 2.0 sites. We also highlight the need to account for automated behavior caused by the technologies used to drive sites such as YouTube when developing models of Web traffic.

A comparison of the results in this thesis to studies performed in parallel to this work was presented. It was found that properties of video files that we observe from the edge network point of view correlate well to observations of file properties made from a more global perspective. We highlight the contrast between local and

global observations of file referencing and summarize key observations from both perspectives. Finally, locality characteristics and transfer properties we observe were compared with results from another study that used measurement from an edge network perspective. We find our results agree well with the results from this parallel study.

Chapter 8

Conclusions and Future Work

This chapter summarizes the thesis and its results. Conclusions and areas for future work are also presented. The content of the thesis is summarized in Section 8.1. Results presented in Chapters 5 and 6 are summarized in Section 8.2. Conclusions and future work are discussed in Sections 8.3 and 8.4, respectively.

8.1 Thesis Summary

This thesis presents an analysis of YouTube usage on the University of Calgary campus over a period of 85 days. Through this study we aimed to get a better understanding of how highly interactive Web sites, like YouTube, are used and how this usage impacts network resources. Data for our analysis comes from two sources; an edge network (the University of Calgary) and the most popular videos lists on YouTube. This combined approach enables us to make some comparisons between videos that are popular locally and videos that are popular globally. We also consider properties of YouTube usage from many different levels including aggregate, user, and session levels. The results of this thesis have many implications for network and service providers who may use this information for capacity planning both at the edge of the network and at central servers.

Chapter 2 presented a primer on relevant background material. The five layer architecture of Internet protocols was presented with a focus on the application and transport layers. At the application layer, detail was provided on the Web including its protocols, evolution, and strategies that have been employed to manage Web traffic. A brief history and introduction to YouTube, the site characterized in this

thesis was also presented.

Related studies relevant to this thesis are presented in Chapter 3. Since YouTube is a Web site centered around delivering multimedia content, related studies include those that consider Web workloads as well as those that consider multimedia workloads. Web workload studies primarily fall into two categories, those that consider general properties of Web traffic (e.g., [6,7,24,33,46]) and those that consider details of user sessions (e.g., [5,8,22,48,50,57]).

Chapter 4 presents the methodology used in this thesis. This thesis uses two main data sets: local data collected from the University of Calgary and global data collected from the list of most popular videos on YouTube. Goals and challenges associated with collecting each of these data sets were discussed as well as the methods we propose to address the challenges.

Results of this thesis are presented in Chapters 5 and 6. They are summarized in the following section.

8.2 Results Summary

Chapters 5 and 6 characterize YouTube usage from an aggregate as well as user and session level, respectively. The main results of these chapters are summarized below:

- *Usage Patterns:* We observe typical campus usage patterns with users being most active during the day on weekdays. We also observe increased popularity of YouTube during our measurement period with a decrease in traffic in mid-February as a result of the reading break.

- *File Sizes:* File sizes of video content are four orders of magnitude larger than the file sizes for the other content types. The average video file size is 10.1 MB with very few extremely large videos. We observe that 90% of the videos have file sizes smaller than 21.9 MB.

- *Video Duration:* Video durations tend to be short with the median video duration observed on campus being 3.33 minutes. Videos on the all time most popular list have less variation in their durations with more than half of the videos having durations between 3 and 5 minutes. We attribute this to the large number of music videos on the all time most popular list.

- *Bit Rate:* Bit rates of YouTube videos on campus are fairly consistent with a mean bit rate of 394 Kbps. This is higher than bit rates observed in previous work [42], likely as a result of increased availability of broadband Internet in recent years.

- *Age of Content:* We observe that videos on the all time most viewed list are older than other videos as a result of the large number of views required to be included in this list. When considering the time since content has been modified, we observe that image and video data is modified less frequently than text or application data. This bodes well for caching image and video content.

- *File Popularity:* Popularity of videos viewed on campus follows a Zipf-like distribution. However, the concentration of references is weak, likely as a result of our edge network point of view. Thus we conclude that the Pareto rule does not hold for videos viewed on campus.

- *Locality Characteristics:* When comparing local and global popularity we find that global popularity of content does not imply local popularity. This suggests that emphasis should be placed on local popularity when determining how to best manage the resource demands of local YouTube usage.

- *Transfer Size:* Similar to file sizes, we observe that the transfer sizes for video objects are orders of magnitude larger than the transfer sizes for other con-

tent types. We also observe distinct spikes in the application transfer sizes corresponding to Javascript files and the Flash video player.

- *Transfer Durations:* Video files have relatively long transfer durations. While other content types have median transfer durations below 1 second, video transfer durations exceed 1 second 96.6% of the time.

- *User Level Interaction:* Variation in transactions and bytes transferred by users can be attributed to varying levels of involvement. Users that view videos tend to have more transactions and transfer significantly more bytes owing to the large size of video content.

- *Session Durations:* Session durations are similar to those observed in previous studies of user sessions [5]. The mean and median session durations are 18.7 and 4.3 minutes, respectively.

- *Inter-transaction times:* We find that inter-transaction times for YouTube are influenced by automated behavior of the browser as a result of Javascripts. This results in a longer threshold being required to distinguish automated requests from user think times.

- *Content Types:* When considering the file types transferred within sessions, surprisingly we notice a large number of sessions that transfer no videos. These sessions without videos are attributed to users that are referred to YouTube content by third party Web sites (e.g., Facebook, MySpace).

8.3 Conclusions

In this thesis, YouTube usage was characterized from both a local and global point of view. We observe campus YouTube usage by monitoring campus Internet traffic

for a period of 85 days during the Winter 2007 semester. Concurrently, information was collected on the most popular videos on YouTube. By observing campus YouTube usage during the semester we were able to observe increasing popularity of YouTube over the course of our measurement period. Collecting information on global YouTube usage also enabled us to determine if videos that were locally popular were also globally popular. Using our data we were able to conduct a thorough analysis of campus YouTube usage and provide implications for network and service providers who will need to provision resources for user generated content based sites in the coming years.

We observe that the multimedia user generated content served by YouTube will challenge conventional content delivery infrastructures. The diverse nature of YouTube users and the content they consume means that caches will require more storage to achieve the same hit rates as they would for traditional Web content. The large file sizes associated with video content compound this problem. While user generated content on sites like YouTube poses many challenges to managing its resource demands, the highly interactive sites that distribute this content provide a wealth of meta-data (e.g., user ratings, view counts etc.) that may be leveraged when developing caching policies for user generated content.

When considering user and session level characteristics of YouTube we observe high variability in user involvement with the site. This can be attributed to some users being referred to YouTube content in the form of embedded videos while visiting other sites. We observe that session durations for campus YouTube users are similar to those observed in previous work. The impacts of the Web technologies used to implement YouTube can be observed in the inter-transaction times of YouTube sessions. Specifically, automated requests that are generated using Javascript can cause longer delays between automated transactions than was previously observed [8].

The measurement methods used in this thesis were designed to overcome the

many challenges of monitoring a Web site that is associated with large volumes of data transfer over an extended period of time. In hindsight, there are many possible enhancements to our measurement approach. First, utilizing a central time stamp on each transaction (from our monitor), rather than relying on the `Date` headers provided by the remote and distributed servers would have provided a finer-grained time stamp, and avoided the accuracy issue we faced with the `Date:` header. Second, we should have recorded additional HTTP headers, to better understand how browser caching affected the workload, as well as how YouTube uses Web caching to minimize the overhead on their delivery infrastructure.

The observations made in this thesis have brought to light some of the challenges that network and service providers will face as highly interactive Web sites like YouTube grow in popularity. Potential avenues for further study of highly interactive Web sites are summarized in the following section.

8.4 Future Work

As discussed in Section 2.2.2, new Web technologies (e.g., AJAX) are enabling Web sites to provide services that they could not provide previously. This has led to many new trends in Web sites including, user generated content, social networks and online software applications. In this thesis, the impacts of multimedia user generated content on network resources were considered. Future directions for study include considering other types of highly interactive Web sites, such as those that focus on social networking (e.g., Facebook, MySpace) or those that feature online software applications. New models of Web traffic may also be developed that leverage results of our YouTube characterization or similar studies.

While this thesis performs one of the first characterizations of user generated content, gathering more data on user generated content would be an interesting

future direction. Multiple data sets would enable any models of user generated content access patterns to be validated to ensure that they hold in general and are not specific to a certain data set. Also, characteristics not considered in this thesis (e.g., spatial locality) could be considered in future studies.

Since social networking Web sites are much more interactive than traditional Web sites it is interesting to observe facets of their usage such as the directionality of data. As users post messages and update their profiles there is the potential for much more traffic directed towards the server than on traditional Web sites. Also, examining the connections between users in social networks and how these connections translate into Web requests can provide insights that may be used to develop new infrastructures to support these sites.

Characterizing usage of online software applications can provide many insights for service providers. As these applications no longer run on the user's computer they must be hosted by the service provider. Understanding the workload characteristics of these online software applications can be used for capacity planning by service providers. Understanding the workload of these online software applications can also be applied by edge network administrators who need to provision their network to meet the resource demands of their users.

A key reason for continuing to perform characterization studies is to ensure that models of Web traffic reflect current characteristics of the Web. The results of this thesis (and future studies) may be applied to update models of Web traffic. Current models of Web traffic have many applications, including simulating background Web traffic in network simulations. Web workload generators (e.g., SPECWeb [61], HTTPerf [38]) also rely on models of Web traffic to evaluate performance of new Web servers.

Bibliography

[1] S. Acharya and B. Smith. An Experiment to Characterize Videos Stored on the Web. In *Proc. of ACM/SPIE Multimedia Computing and Networking (MMCN) '98*, San Jose, USA, January 1998.

[2] S. Acharya, B. Smith, and P. Parnes. Characterizing User Access to Videos on the World Wide Web. In *Proc. of ACM/SPIE MMCN '00*, San Jose, USA, January 2000.

[3] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control . RFC 2581 (Proposed Standard), April 1999. Updated by RFC 3390.

[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. Arlitt. Characterizing Web User Sessions. *ACM SIGMETRICS Performance Evaluation Review*, 28(2):50–56, 2000.

[6] M. Arlitt and T. Jin. Workload Characterization of the 1998 World Cup Website. *IEEE Network*, 14(3):30–37, 2000.

[7] M. Arlitt and C. Williamson. Internet Web Servers: Workload Characterization and Performance Implications. *IEEE/ACM Transactions on Networking*, 5(5):631–645, 1997.

[8] P. Barford and M. Crovella. Generating Representative Web Workloads for Network and Server Performance Evaluation. In *Proc. of ACM SIGMETRICS '98*, Madison, USA, June 1998.

[9] Tim Berners-Lee. Information Management: A Proposal. `http://www.w3.org/History/1989/proposal.html`, March 1989.

[10] Blogspot. `http://www.blogspot.com`.

[11] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proc. of IEEE INFOCOM*, New York, USA, March 1999.

[12] Bro. `http://www.bro-ids.org`.

[13] R. Bunt and J. Murphy. The Measurement of Locality and the Behaviour of Programs. *Computer Journal*, 27(3):238–253, 1984.

[14] CBC. YouTube's Bride Wig Out Revealed as 'net seed' for Ad Campaign. *CBC Arts*, February 2007.

[15] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *Proc. of ACM SIGCOMM Internet Measurement Conference (IMC) '07*, San Diego, USA, October 2007.

[16] 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.

[17] L. Cherkasova and M. Gupta. Analysis of Enterprise Media Server Workloads: Access Patterns, Locality, Content Evolution, and Rate of Change. *IEEE/ACM Transactions on Networking*, 12(5):781–794, 2004.

[18] M. Chesire, A. Wolman, G. Voelker, and H. Levy. Measurement and Analysis of a Streaming Media Workload. In *Proc. of USENIX Symposium on Internet*

Technologies and Systems (USITS) '01, San Francisco, USA, March 2001.

[19] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. Almeida, and B. Ribeiro-Neto. Analyzing Client Interactivity in Streaming Media. In *Proc. of WWW '04*, New York, USA, May 2004.

[20] M. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[21] Mark Crovella and Balachander Krishnamurthy. *Internet Measurement: Infrastructure, Traffic and Applications*. John Wiley and Sons Ltd., 2006.

[22] C. Cunha, A. Bestavros, and M. Crovella. Characteristics of World Wide Web Client-based Traces. Technical Report BUCS-TR-1995-010, Boston University, CS Dept, Boston, MA 02215, April 1995.

[23] P. Denning. Working Sets Past and Present. *IEEE Transactions on Software Engineering*, 6(1):64–84, 1980.

[24] B. Duska, D. Marwood, and M. Freeley. The Measured Access Characteristics of World-Wide-Web Client Proxy Caches. In *Proc. of USITS '97*, Monterey, USA, March 1997.

[25] Facebook. `http://www.facebook.com`.

[26] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Proposed Standard), June 1999.

[27] Flickr. `http://www.flickr.com`.

[28] S. Floyd, M. Handley, J. Padhye, and J. Widmer. Equation-Based Congestion Control for Unicast Applications . In *Proc. of ACM SIGCOMM '00*, Stockholm, Sweden, August 2000.

[29] S. Floyd, T. Henderson, and A. Gurtov. The NewReno Modification to TCP's Fast Recovery Algorithm. RFC 3782 (Proposed Standard), April 2004.

[30] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube Traffic Characterization: A View From the Edge. In *Proc. of ACM SIGCOMM IMC '07*, San Diego, USA, October 2007.

[31] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Characterizing User Sessions on YouTube. In *Proc. of ACM/SPIE MMCN '08*, San Jose, USA, January 2008.

[32] Google Docs. `http://docs.google.com`.

[33] S. Gribble and E. Brewer. System Design Issues for Internet Middleware Services: Deductions from a Large Client Trace. In *Proc. of USITS '97*, Monterey, USA, March 1997.

[34] K. Gummandi, R. Dunn, S. Saroiu, S. Gribble, H. Levy, and J. Zahorjan. Measurement, Modeling and Analysis of a Peer-to-Peer File-Sharing Workload. In *Proc. of ACM Symposium on Operating Systems Principles (SOSP)*, Bolton Landing, USA, October 2003.

[35] 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. of ACM SIGCOMM IMC '06*, Rio de Janeiro, Brazil, October 2006.

[36] M. Halvey and M. Keane. Exploring Social Dynamics in Online Media Sharing. In *Proc. of of WWW '07*, Banff, Canada, May 2007.

[37] 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, October 1999.

[38] Httperf. `http://www.hpl.hp.com/research/linux/httperf/`.

[39] V. Jacobson. Congestion avoidance and control. In *Proc. of ACM SIGCOMM '88*, New York, USA, August 1988.

[40] J. Kurose and K. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison-Wesley, 2005.

[41] S. Lee, P. Kim, and H. Jeong. Statistical Properties of Sampled Networks. *Physical Review E*, 73(016102), 2006.

[42] M. Li, M. Claypool, R. Kinicki, and J. Nichols. Characteristics of Streaming Media Stored on the Web. *ACM Transactions on Internet Technology*, 5(4):601–626, 2005.

[43] Business Intelligence Lowdown. Top 10 Largest Databases in the World. `http://www.businessintelligencelowdown.com/2007/02/top_10_largest_.html`, February 2007.

[44] A. Mahanti, D. Eager, M. Vernon, and D. Sundaram-Stukel. Scalable On-demand Media Streaming with Packet Loss Recovery. *IEEE/ACM Transactions on Networking*, 11(2):195–209, 2003.

[45] A. Mahanti, D. Eager, and C. Williamson. Temporal Locality and its Impact on Web Proxy Cache Performance. *Performance Evaluation Journal (Special Issue on Internet Performance Modelling)*, 42(2-3):187–203, 2000.

[46] A. Mahanti, C. Williamson, and D. Eager. Traffic Analysis of a Web Proxy Caching Hierarchy. *IEEE Network*, 14(3):16–23, 2000.

[47] S. Majumdar and R. Bunt. Measurement and Analysis of Locality Phases in File Referencing Behaviour. In *Proc. of ACM SIGMETRICS '86*, Raleigh, USA, June 1986.

[48] H. Marques, L. Rocha, P. Guerra, J. Almeida, Jr. W. Meira, and V. Almeida. Characterizing Broadband User Behavior. In *Proc. of the ACM workshop on Next-generation Residential Broadband Challenges (NRBC)*, New York, USA, October 2004.

[49] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgement Options. RFC 2018 (Proposed Standard), October 1996.

[50] D. Menascé, V. Almeida, R. Fonseca, and M. Mendes. A Methodology for Workload Characterization of E-commerce Sites. In *Proc. of the 1st ACM conference on Electronic commerce*, Denver, United States, November 1999.

[51] J. Milani. Coming to Your Screen: DIY TV. *BBC Money Programme*, 2007.

[52] P. Mockapetris. Domain Names - Concepts and Facilities. RFC 1034 (Proposed Standard), November 1987.

[53] P. Mockapetris. Domain Names - Implementation and Specification. RFC 1035 (Proposed Standard), November 1987.

[54] M. Musgrove. Viacom Decides YouTube Is a Foe. *Washington Post*, February 2007.

[55] My Space. `http://www.myspace.com`.

[56] University of Calgary. About the U of C. `http://ucalgary.ca/about/`.

[57] Adeniyi Oke and Richard B. Bunt. Hierarchical Workload Characterization for a Busy Web Server. In *Proc. of the 12th International Conference on Computer Performance Evaluation, Modelling Techniques and Tools (TOOLS '02)*, London, UK, April 2002.

[58] T. O'Reilly. What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software. `http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html`, September 2005.

[59] V. Paxson. Empirically-Derived Analytic Models of Wide-Area TCP Connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, 1994.

[60] J. Postel. User Datagram Protocol. RFC 768 (Proposed Standard), August 1980.

[61] Standard Performance Evaluation Corporation (SPEC). Web Servers. `http://www.spec.org/benchmarks.html#web`, 2005.

[62] K. Sripanidkulchai, B. Maggs, and H. Zhang. An Analysis of Live Streaming Workloads on the Internet. In *Proc. of ACM SIGCOMM IMC '04*, Taormina, Italy, October 2004.

[63] Sustainability of Digital Formats Planning for Library of Congress Collections. Macromedia Flash FLV Video File Format. `http://www.digitalpreservation.gov/formats/fdd/fdd000131.shtml#specs`, March 2007.

[64] Tcpdump. `http://www.tcpdump.org`.

[65] USA Today. YouTube Serves up 100 million Videos a Day Online. `http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm?`, July 2006.

[66] A. Williams, M. Arlitt, C. Williamson, and K. Barker. Web Workload Characterization: Ten Years Later. *Web Content Delivery*, pages 3–21, 2005.

[67] Wordpress. `http://www.wordpress.com`.

[68] YouTube. `http://www.youtube.com`.

[69] H. Yu, D. Zheng, B. Zhao, and W. Zheng. Understanding User Behavior in Large-Scale Video-on-Demand Systems. *ACM SIGOPS Operating Systems Review*, 40(4):333–344, 2006.

[70] Q. Zhang, W. Zhu, and Y. Zhang. Network-Adaptive Rate Control and Unequal Loss Protection with TCP-Friendly Protocol for Scalable Video Over Internet. *VLSI Signal Processing Systems*, 34(1-2):67–81, 2003.

[71] M. Zink, K. Suh, and J. Kurose. Watch Global, Cache Local: YouTube Network Traffic at a Campus Network - Measurement and Implications. In *Proc. of ACM/SPIE MMCN '08*, San Jose, USA, January 2008.

[72] G. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley (Reading MA), 1949.