Computer Networks 53 (2009) 501–514

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Characteristics of YouTube network traffic at a campus network – Measurements, models, and implications

Michael Zink [a,*], Kyoungwon Suh [b], Yu Gu [a], Jim Kurose [a]

[a] Department of Computer Science, University of Massachusetts, Amherst, MA 01003, United States
[b] School of Information Technology, Illinois State University, Normal, IL 61790, United States

ARTICLE INFO

Article history:
Available online 5 November 2008

Keywords:
Measurement study
Content distribution
Caching
Peer-to-Peer

ABSTRACT

User-Generated Content has become very popular since new web services such as YouTube allow for the distribution of user-produced media content. YouTube-like services are different from existing traditional VoD services in that the service provider has only limited control over the creation of new content. We analyze how content distribution in YouTube is realized and then conduct a measurement study of YouTube traffic in a large university campus network. Based on these measurements, we analyzed the duration and the data rate of streaming sessions, the popularity of videos, and access patterns for video clips from the clients in the campus network. The analysis of the traffic shows that trace statistics are relatively stable over short-term periods while long-term trends can be observed. We demonstrate how synthetic traces can be generated from the measured traces and show how these synthetic traces can be used as inputs to trace-driven simulations. We also analyze the benefits of alternative distribution infrastructures to improve the performance of a YouTube-like VoD service. The results of these simulations show that P2P-based distribution and proxy caching can reduce network traffic significantly and allow for faster access to video clips.

1. Introduction

During the past two years YouTube has become a very popular web application. Viewers around the world request millions of videos every day.[1] So far, there is no indication that this popularity will decrease and indeed is more likely to increase, with video clips from broadcasters and media companies being streamed not only to desktop PCs but also to mobile devices.

YouTube represents a service that is different from the traditional VoD systems in two important respects. First, traditional VoD systems usually offer professionally-produced video content such as movies, news, sport events or TV series. The quality and popularity of these contents are well-controlled and predictable. In contrast, YouTube videos can be uploaded by anyone with access to the Internet. The content and quality of these video clips vary significantly. Consequently, predicting how many new videos will be uploaded and the popularity of the videos is much more difficult. Second, the manner in which content is distributed in traditional VoD systems differs from the distribution used for YouTube. In a VoD system, either the viewers expect regular updates on the content (in the case of news or TV series[2]) or the content provider announces new content (e.g., the release of a latest block buster). In the YouTube system, it is often the case that a video clip becomes extremely popular after viewers become aware of the clip and tell their friends about it, discuss it in BLOGs,

* Corresponding author. Tel.: +1 413 545 4465.
E-mail address: zink@cs.umass.edu (M. Zink).

[1] http://www.usatoday.com/tech/news/2006-07-16-youtube-views_x.htm http://en.wikipedia.org/wiki/YouTube.

[2] Many TV channels already make a large portion of their TV series available via their web sites after they have been originally broadcast on traditional TV. In this case, viewers know when a new sequel will be available.

1389-1286/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2008.09.022

and put embedded links to the clip on their own web pages. Thus, YouTube is truly a community site that allows users to upload and share their videos.

The objective of our work is to investigate the relationship between the local and global popularity and the impact of time scale of measurement and user population on the popularity of YouTube video clips in a campus network. We also use information gained from a statistical analysis of the trace data to generate new (synthetic) traces. The synthetic traces can be used as input to simulations that could, for example, show how various distribution infrastructures perform when the viewer population and the request arrival rates for YouTube videos grow in the near future. We describe how we can create such synthetic traces and present the performance of distribution infrastructures based on simulations using both real and synthetic traces as input data. Our analysis builds on our previous short-term measurement study of YouTube traffic presented in [1]. In our previous work, we focused on how the performance of various content distribution architectures can be affected by the characteristics of user requests for YouTube videos. In an effort to extend this previous research, we also investigate the focus in this article on the longer-term behavior of user requests from a campus network. In addition, we develop models of YouTube requests based on statistical information obtained from the measurements presented in this article. To investigate the long-term characteristics of user requests, we conducted additional measurements that include a set of six measurement traces (three existing ones [1] and three new ones) spanning a period of 10 months. Four of these traces are used for further analysis in this article.

The study in our campus network shows that trace statistics are relatively stable over a short-term measurement period while long-term trends can be observed. In general, our observations are consistent with the results presented in related work [2,3] analyzing the YouTube service. One interesting long-term observation from our measurement study is an approximately 10% increase in the number of videos requested multiple times from clients in our campus network. This result implies that a distribution infrastructure such as proxy caching would become more effective in the long-term, if applied to YouTube content distribution. A more in-depth investigation of the correlation between the local and global popularity is consistent with our finding in [1] that there is no strong correlation between local and global popularity. In addition, we show how synthetically generated data can be used to analyze the benefit of alternative distribution methods such as P2P and proxy caching under higher loads than the ones observed in our measurement data.

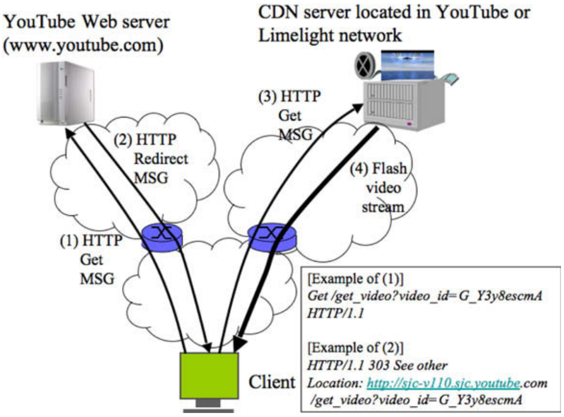
The rest of the article is structured as follows. Section 2 presents the basic interaction between a YouTube server and a client browser and describes the monitoring environment and procedure for YouTube traffic measurements. An in-depth analysis of the trace data is presented in Section 3. Section 4 describes a method to generate synthetic traces based on the measured traces. The evaluation of these synthetic traces and the performance of P2P-based and proxy caching based on these traces is described in Section 5. Section 6 presents related work. Finally, we conclude the article in Section 7.

2. YouTube functionality and traffic monitoring

In this section, we briefly overview how a client retrieves a video clip from a YouTube server and describe our monitoring process of YouTube signaling traffic between our campus network clients and YouTube servers. A more detailed description on the monitoring process, including a description on how the actual video stream can be monitored, can be found in [4,1].

2.1. How YouTube works

YouTube is a web-based service that provides video sharing via the Internet. Clients can upload their own videos and make them available to the public. Viewers can search for videos and then watch these videos on their computers or mobile devices. This section gives an overview on how the YouTube system works. We describe this procedure from the perspective of a client who requests a video from the YouTube web site with an emphasis on the communications between the client and YouTube servers. Fig. 1 illustrates the communication among the client, YouTube server, and a server of the Content Distribution Network (CDN). When a client has chosen a specific video, an HTTP GET message is sent from the client to the YouTube web server. This message indicates that the client is requesting a specific video which is identified by a unique video identifier, in this example *G_Y3y8escmA*. After receiving the GET message, the web server replies with an HTTP 303 See Other message, namely the redirection message. This message contains a location response-header field, which redirects the client to the video server from which the video will be streamed. An example of HTTP GET and HTTP 303 See Other messages is shown in Fig. 1. The redirection mechanism introduces load balancing to the system since the web server can redirect the client to a video server in a dynamic way. Therefore, the web server must know which videos can be served from which video servers and the actual load of the video servers. To allow



[Example of (1)]
Get /get_video?video_id=G_Y3y8escmA
HTTP/1.1

[Example of (2)]
HTTP/1.1 303 See other
Location: http://sjc-v110.sjc.youtube.com
/get_video?video_id=G_Y3y8escmA

Fig. 1. Video retrieval in YouTube.

for load balancing, YouTube makes use of a CDN service provided by both the YouTube and Google Video server farms and the Limelight CDN. Such a service automatically distributes content (in this case videos) to servers in the CDN. The objective of our work is not to identify the content distribution mechanism used by the CDN provider,[3] but rather to analyze the impact of YouTube traffic on *local* distribution infrastructures based on the results of our measurement study and to gather statistical information that can be used to generate YouTube traffic models with a focus on local access networks.

2.2. Monitoring YouTube traffic

In this section, we describe our methodology for monitoring signaling and data traffic between clients in our campus network and YouTube servers. The methodology allows us to understand how a client retrieves a video stream from YouTube; we then obtain YouTube usage statistics in our campus network. Monitoring is a two-step process. In the first step, we analyze signaling traffic, i.e., HTTP message exchange between the client and the YouTube web server. Following the first step, we monitor video streaming traffic, i.e., the TCP headers of the video data sent from the CDN video servers.

2.2.1. The monitoring environment

We use the following measurement equipment to collect traces of the traffic between clients in our campus network and YouTube servers. The measurement equipment is a commodity PC, with an installed Data Acquisition and Generation (DAG) card [5] to capture packet headers, placed at the gateway router of UMass Amherst, and connected via optical splitters to the Giga-bit access links connecting the campus network to a commercial network. The TCP and IP headers of all the packets that traverse this link are captured by the DAG card, along with the current timestamp. In addition, we capture the HTTP headers of all the HTTP packets coming from www.youtube.com. Note that all recorded IP addresses are anonymized.[4]

2.2.2. Monitoring web server access

The arrival time, size, source and destination IP addresses, and port numbers are recorded for each outgoing packet through the gateway router. If the destination or source IP address belongs to the pool of YouTube web servers, the HTTP header is additionally recorded. Initially, a client has to connect to the YouTube web site to search and then watch a video. The recorded information captures both the HTTP GET message and the HTTP 303 See Other message as described in the previous section. These messages allow us to extract details of the requested video, such as video identifier, when it is requested, and from which CDN server the data will be streamed. By tracing the video identifier, we can determine the frequency with which a certain video is requested from clients in our campus network.

3. Measurement results

In this section, we analyze the YouTube traces obtained as described in Section 2.2. The main goal of this analysis is two-fold. The first goal is to investigate the local popularity of YouTube video clips, the relationship between local and global popularity, and how time scale of the measurement and user population impact local popularity. Local popularity defines the popularity among the video clips requested in a trace file. Global popularity on the other hand defines the popularity of the locally requested video clips based on popularity information given at YouTube.com. (A more detailed description of global popularity is given in Section 3.2.) The second goal is to gather empirical information from the traces that will be used to generate modeled data for a variety of simulations, as discussed in Sections 4 and 5. In the remainder of this section, we describe the different traces that we have collected, analyze the local popularity distribution and compare it with the global popularity distribution. Furthermore, we investigate the influence of time scale on popularity, and investigate the popularity for individual clients and the long-term trends on these parameters. In addition, we present the results from a transport-level traffic analysis of the actual video data transmission.

3.1. Traces

This section describes the six traces that were collected for the measurement study presented in this article. Our goal here is to extend the measurement set we have collected in [1]. The traces presented in [1] were used to investigate the effect of (i) trace length, (ii) user population, and (iii) content of local importance on the performance of different distribution architectures. The new set of three traces presented in this article were collected to additionally investigate characteristics of YouTube traffic in a campus network over a longer time span and also to evaluate the correlation between local and global popularity in more detail.[5]

Table 1 gives an overview of our measurement traces. The traces represent data from 12-h to two-week long measurements. The six traces span a period of 10 months with the first one taken in May, 2007 and the last taken in March, 2008. This last trace serves as data to understand the long-term effects of YouTube traffic and also to more accurately estimate the correlation between local and global popularity of YouTube video clips. "Per video stats" rows in Table 1 contain statistics about the distinct videos requested during the monitoring period for each trace. The "Total" row under "Per video stats" shows the total number of distinct videos requested during the trace period; "Single" shows the percentage of videos requested only once; and "Multi" indicates the percentage of videos

[3] Detailed information about the content management can be found at http://www.limelightnetworks.com

[4] We limit the scope of monitoring to such types of payloads, because of the privacy issue and the limited computational and I/O capability of the monitoring equipment to capture headers at giga-bit speed.

[5] All traces presented in this article can be found at http://traces.cs.umass.edu/index.php/Network/Network.

Table 1
YouTube traffic traces.

Trace		T1	T2	T3	T4	T5	T6
Date		05/08/07–05/09/07	05/22/07–05/25/07	06/02/07–06/06/07	09/04/07–09/11/07	01/29/08–02/12/08	03/11/08–03/18/08
Length (h)		12	72	108	162	336	168
# Of unique clients		2127	2480	1547	7538	16336	8879
Per video stats	Total	12955	23515	17183	82132	303331	131450
	Single (%)	77.4	77.7	77.1	72.5	65.9	68.5
	Multi (%)	22.6	22.3	22.9	27.5	34.1	31.5
Request stats	Total	18040	32971	24211	145140	611968	243023
	Single (%)	55.6	55.3	54.7	41.1	32.7	37.1
	Multi (%)	44.4	44.7	45.3	58.9	67.3	62.9

requested more than once. The "Request stats" rows present statistics of the requests by clients in the campus network. "Request stats" differs from "Per video stats" in that the former presents a client-centric view of the measurement traces. The "Total" row under "requests stats" shows the total number of requests made by clients in the campus network during the measurement period. "Single" shows the percentage of requests from clients that make only a single YouTube request during the measurement and "Multi" shows the percentage of requests from clients that demand multiple YouTube videos.

- T1 presents measurement results from a short duration. This trace is used to investigate short-term behavior with respect to request rate and popularity of video clips.
- T2 was collected during the last week of spring 2007 semester. Due to the larger population of students on campus, we expected a larger number of overall requests during the measurement period than during a measurement that was executed during a break period.
- T3 was collected during the first week of the summer break and we expected a smaller number of overall requests because of the smaller student population on campus. The results shown in the *Total requests* row of Table 1 confirm this conjecture. This trace allows us to investigate the influence of different user populations (in comparison to the other five measurements) on request rate and video clip popularity. T3 also allows us to investigate whether video content that is of local importance can have an influence on popularity and request rate. Shortly before that measurement was conducted, a video clip showing part of the University's commencement was published on YouTube. Not surprisingly, this clip had the second highest popularity of all the clips requested during this measurement.
- The trace collection of T4 intentionally coincides with the beginning of the Fall 2007 semester. For this trace, we expected a large number of overall requests because of the sharp increase in the number of students returning back to campus. The conjecture is confirmed by the result shown in the Total requests row that shows a significantly larger value than the one for T3. Since the measurement periods for both traces are not equal, we decided to compare the 24-h request rate instead. The 24-h rate for T4 is almost 80% higher than the one for T3 (see 8th row in Table 2).
- Similar to T4, the beginning of the measurement for *T5* coincides with the beginning of the Spring 2008 semester. This measurement period was chosen to have comparable conditions as for the T4 measurement. Under these comparable conditions we investigate long-term trends among traces.
- T6 is a *composite* trace that is composed of seven consecutive 24-h traces. This composite trace allows us to investigate the correlation between local and global popularity in more detail than in [1]. (The collection of this trace is described in more detail in Section 3.2.)

The analysis of the first three traces, i.e., T1, T2 and T3 has been reported in our previous work [1] and consequently we will focus on traces T2, T4, T5 and T6 in more detail in this article. We decided to focus on these four traces since they were all taken during the spring or fall semester with a high student population on campus. Compared to the results we presented in [1] where the data for "Per video

Table 2
YouTube normalized traffic traces.

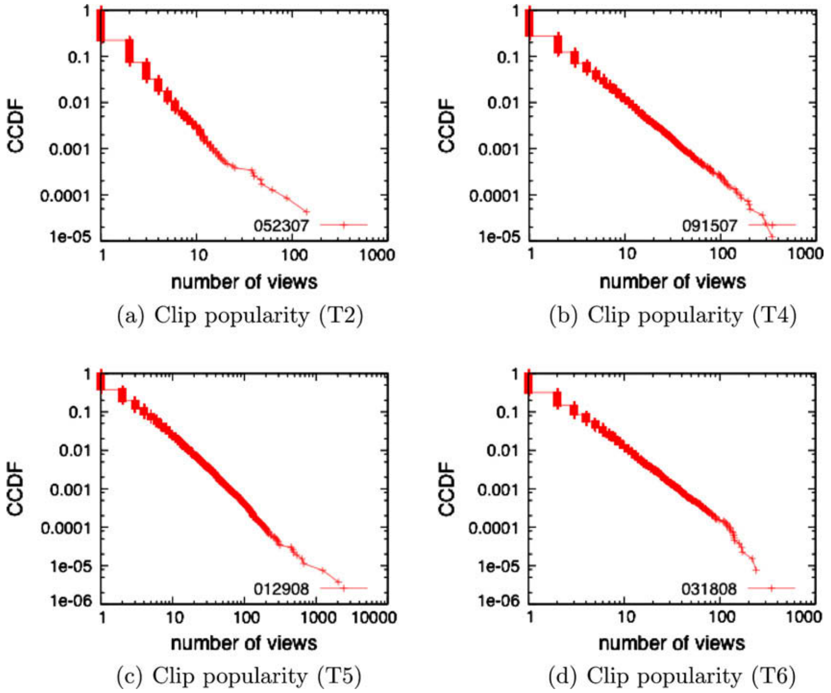
Trace		N2	N3	N4	N5	N6
Date		05/23/07	06/06/07	09/05/07	02/06/08	03/12/08
Length (h)		24	24	24	24	24
# Of unique clients		1403	597	2376	4555	5180
Per video stats	Total	10201	4512	13500	28782	37743
	Single (%)	79.8	81.4	80.5	72.4	72.9
	Multi (%)	20.2	18.6	19.5	27.6	27.1
Request stats	Total	13511	5771	18750	46407	61140
	Single (%)	60.3	63.7	58.0	44.9	45.1
	Multi (%)	39.7	36.3	42.0	55.1	54.9

stats" and "Request stats" are quite similar, the results in this article show a greater variation. Since the duration for all traces is not identical, it cannot be determined exactly if trace duration or long-term trends are the primary cause of these variations. Therefore, we decided to look at a 24-h snapshot from T2, T4, T5, and T6 instead. In addition, we chose an identical day of the week (Wednesday) from these four data sets. Results from this snapshot data for all four traces, denoted Traces N2, N4, N5, and N6 are shown in Table 2. For completeness, we also show the 24-h snapshot of T3, i.e., N3. A 24 h snapshot for T1 could not be created since it is only 12 h long. This data snapshot allows us to analyze long-term trends without the bias potentially caused by different-lengths traces. The results from Table 2 show the interesting fact that the trace data is relatively stable over shorter periods while it changes in the longer-term. There is only a small change in the data between Trace N2 and N4 on one hand and Trace N5 and N6 on the other. But there are significant changes between traces that are separated further in time (i.e., Trace N4 and N5). For example, the statistics for *Multi* and *Single* are quite similar between trace pairs N2, N4 and N5, N6 while there is a significant difference between the two groups.

3.2. Popularity of YouTube video clips

One of the design criteria for video distribution systems is the popularity distribution of video clips requested by users in the access network. Obtaining and then analyzing video clip popularity allows us to identify critical design parameters for the appropriate infrastructure for a video distribution system like YouTube. For example, local caching implemented in access networks can reduce the amount of network traffic if a small subset of the requested videos have high popularity. In addition, the popularity distribution can be used to derive modeled data from the traces. In Section 4, we use the popularity distribution to generate synthetic YouTube traces. Fig. 2 shows the complementary cumulative distribution function (CCDF) of the popularity (number of times a video is viewed) of video clips for traces T2, T4, T5, and T6. The CCDF of a random variable X is defined as $\overline{F}_X(x) = P(X \geqslant x)$. The x-axis shows the number of requests for video clips and the y-axis shows the CCDF in log scale. We observe that the popularity distribution is relatively similar among the four traces.

The results in Fig. 2 also show that a large number of video clips are requested only once during the measurement period (see also Tables 1 and 2). To get a better insight into the number of requests per video clip we determine the rate between the number of video clips with x requests and the total number of requested video clips and show the results in Fig. 3. The leftmost bar of each graph in Fig. 3 shows the percentage of video clips that were requested only once during the measurement period. The second bar shows the percentage of video clips that were requested twice, and so on; a gap along the x-axis indicates that no video clips were requested the corresponding number of times represented by the x-value. Clearly, the number of video clips with two or more requests will increase as the measurement period gets longer. Therefore, we focus on the 24-h snapshot results (shown in Table 2) to analyze long-term trends in our campus network, and to analyze the influence of client population on the popularity of video clips. By comparing the data shown in the "Per video stats" row, we can observe a long-term trend that the number of video clips requested only once in a 24-h period



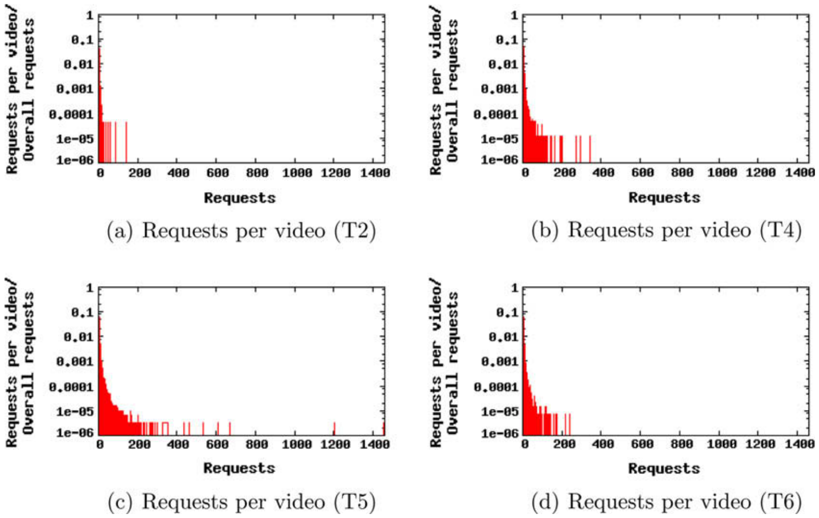
(a) Clip popularity (T2)

(b) Clip popularity (T4)

(c) Clip popularity (T5)

(d) Clip popularity (T6)

Fig. 2. Video popularity for all four traces. *X*-axis represents the number of requests and *Y*-axis is the CCDF in log scale.



(a) Requests per video (T2)

(b) Requests per video (T4)

(c) Requests per video (T5)

(d) Requests per video (T6)

Fig. 3. Percentage of requests per video versus overall requests.

decreases and that the number of clips that are requested at least twice increases. If this trend continues to hold, local caching mechanisms described in Section 5 can make YouTube more efficient locally.

In general, the benefit gained by local caching can be increased if popularity information about each video content is available. Unfortunately, this popularity information is neither available nor easy to obtain. However, for YouTube videos, it is possible to obtain some popularity information. For each video clip the number of views is given on the YouTube web site.

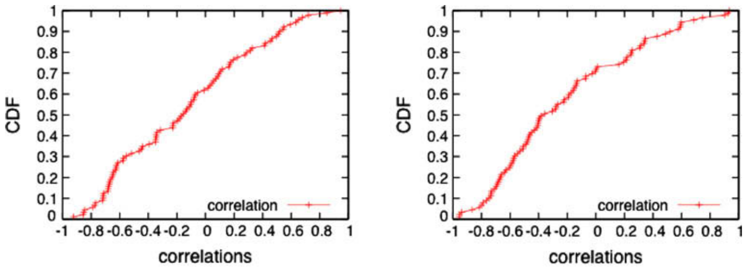
In the present work, we refine our previous work [1] by conducting a series of seven day measurements of 24-h traces. From the trace data, we obtained the global popularity information of the requested video clips immediately after the end of the measurement. For each video clip that appears in all seven traces, we obtained its daily local popularity x_i, $i = 1, \ldots, 7$ by counting its appearance in the trace. We also calculated its daily global popularity y_i, $i = 1, \ldots, 7$ from the global popularity obtained after each measurement. The global popularity was obtained by retrieving the ‘‘views" information from YouTube.com and the according popularity ranking for these specific video clips. E.g., if video clip A had 100 views and video clip B had 50 views, then A was ranked higher than B. We then used this information to analyze the correlation of daily local and global popularity. The correlation coefficient between local and global popularity is calculated as:

$$\rho_{X,Y} = \frac{\sum_{i=1,\ldots,7}((x_i - \overline{X})(y_i - \overline{Y}))}{\sqrt{\sum_{i=1,\ldots,7}(x_i - \overline{X})^2 \sum_{i=1,\ldots,7}(y_i - \overline{Y})^2}},$$

where X is the vector of the number of local views obtained from our gateway traces, Y is that obtained from the YouTube web site, and $\bar{X}$ is the sample mean of X. The correlation coefficients for all those video clips that appear in all seven traces are shown in Fig. 4. As can be seen from this figure, there is only a small set of video clips which have a high correlation between local and global popularity, confirming our findings in [1]. We also analyzed the correlation between local and global popularity where the local data is offset by a day, i.e, the global popularity of day k is compared with the local popularity of day $k + 1$. This analysis was motivated by the fact that YouTube promotes video clips on its home page and this might lead to a delayed local popularity of these video clips. Also in this case only a small set of video clips show a high correlation between local and global popularity, as shown in Fig. 4b. Evidently, the promotion of video clips on the YouTube home page does not strongly influence local popularity within our campus.

3.3. Request rate over time

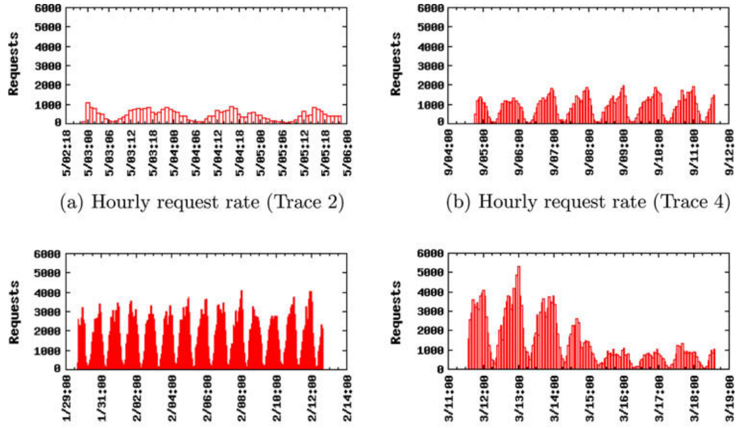
In this section, we analyze the request rate for YouTube video clips over time. First we examine the binned hourly request rate for all video clips, as shown in Fig. 5. The figure shows the expected result that the request rate is high during the day and the evening until around midnight and then decreases significantly in the early morning hours. The request rate also drops slightly in the early evening hours (around 18:00) and increases again later in the evening (20:00). We conjecture that this effect is caused by the fact that the majority of students go for dinner during that time span. The subfigures show a long-term trend of YouTube requests. While for T2 the peak request rate reaches 1000 requests per hour, the rate increases to more than 3000 requests per hour and reaches up to 4000 requests per hour in T6 (before the break starts). We conjecture from these observation that, in the long-term, the request rate for YouTube video clips from our campus network is constantly increasing. Another interesting effect that can be observed in this data is the drop in request rate in T6: Fig. 5d shows that the start of spring break (3/15/08) leads to a sharp decline in the overall request rate by more than 50%.



(a) Distribution of same-day correlation between local and global popularity

(b) Distribution of next-day correlation between local and global popularity

Fig. 4. Correlations between local and global popularity.



(a) Hourly request rate (Trace 2)

(b) Hourly request rate (Trace 4)

(c) Hourly request rate (Trace 5)

(d) Hourly request rate (Trace 6)

Fig. 5. Hourly request rate for video clips from the traces shown in Table 1.

3.4. Request inter-arrival time

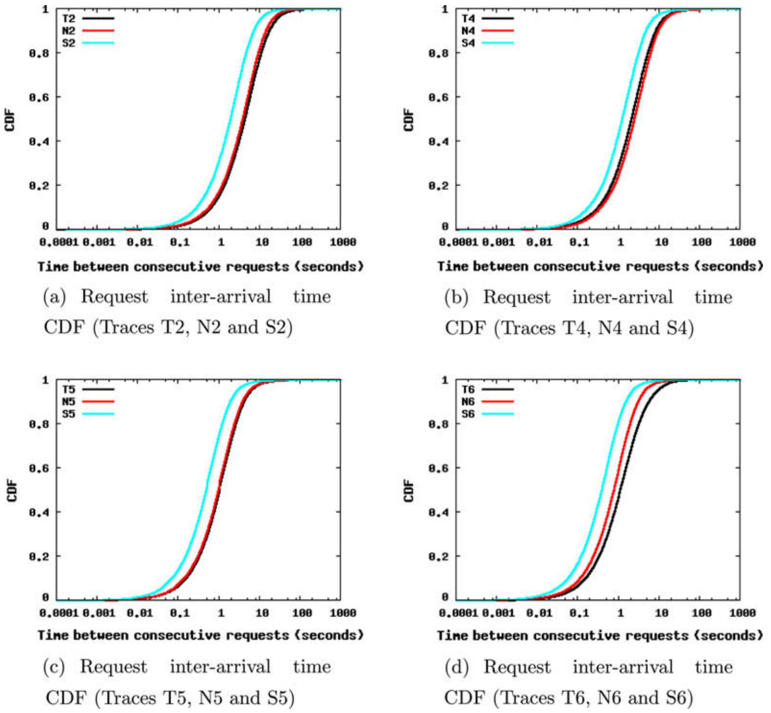
The analysis of the request inter-arrival time allows us to obtain the overall traffic load placed on our campus network by YouTube traffic. The analysis can also suggest which alternative caching and distribution architecture are appropriate, as discussed in Section 5. The results presented here can also be used to model the inter-arrival time for YouTube video requests. Fig. 6 shows the CDF of the time interval between two consecutive YouTube video requests from our campus network. The median inter-arrival times for traces T2, T4, T5, and T6 are 4.3, 2.1, 1, and 1.1 s, respectively. Compared to the results on video popularity (see Section 3.2), the inter-arrival time shows an interesting long-term trend, with median inter-arrival times decreasing from 4.3 to 1 second during the 10 month period. Fig. 6 also shows the CDF of the inter-arrival time for the 24-h snapshot traces N2, N4, N5, and N6 which are almost identical to the CDF of traces T2, T4, T5, and T6. The only pair of traces showing some difference is T6 and N6 (Fig. 5d). Here trace T6 shows a higher percentage of longer inter-arrivals than the snapshot trace N6. This could be caused by the fact that part of T6 was collected during spring break with a much smaller student population at the end of the trace collection period. The data we chose for trace N6 was a day before spring break with a large student population. The results for traces S2, S4, S5, and S6 will be further discussed in Section 4.3.

3.5. Requests per client

We now examine the distribution of requests per client, indicating the extent to which clients sent multiple requests, and how many clients send multiple requests for a given video clip in a certain time period, and if such requests come from a large group of clients or a small group of clients. In addition, this information is used to model YouTube traffic, as described in Section 4.

Table 1 shows that requests were made from 2480, 7538, 16336, and 8879 unique clients for Traces T2, T4, T5 and T6, respectively.[6] Due to the difference in measurement periods for these four traces we will from here on focus on the 24-h snapshot traces to better compare requests for video clips made per client. The number of unique clients

[6] The actual number of physical clients might be slightly higher due to the usage of DHCP for some IP addresses in our campus network.



(a) Request inter-arrival time CDF (Traces T2, N2 and S2)

(b) Request inter-arrival time CDF (Traces T4, N4 and S4)

(c) Request inter-arrival time CDF (Traces T5, N5 and S5)

(d) Request inter-arrival time CDF (Traces T6, N6 and S6)

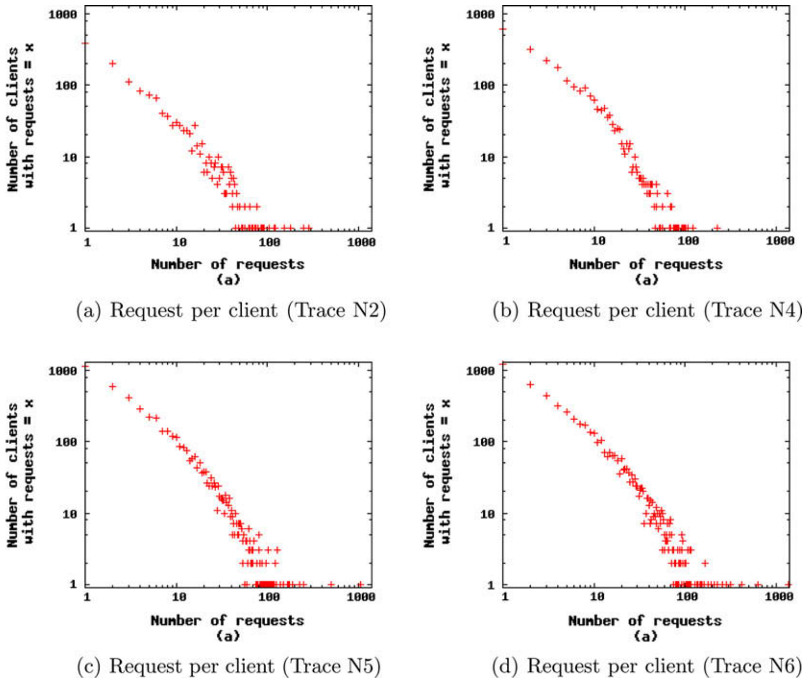
Fig. 6. CDF of the request inter-arrival times for traces T2, T4, T5, T6, N2, N4, N5, N6, S2, S4, S5, and S6.

per 24-h interval are shown in row 4 of Table 2. Not surprisingly, the number of clients is the smallest in Trace N2 with a value of 1403 due to the fact that this measurement was collected during spring 2007 semester. For the measurements taken during the fall 2007 and spring 2008 semester, a significant increase in the number of clients is shown. The number of clients has almost doubled (from 1765 to 3397) over the period of 5 months. This is consistent with the increase in other statistics shown in Table 2. The scatter plots in Fig. 7 shows the distribution of number of requests over the overall client population. On the x-axis the number of requests per client is shown and the y-axis shows the number of clients with x requests. For example, Fig. 7a shows 382 clients have issued only one request during the measurement period for Trace N2. All four graphs show a significant number of clients that issue two or more requests for a video clip.

3.6. *Video clip traffic analysis*

In addition to the analysis of HTML-based control message exchange, we also analyze the transport-level messages that carry the video clip data from the content servers to destinations in the campus network. Due to the asymmetric routing caused by the addition of a link to a new ISP from our campus in the fall of 2007, we were only able to monitor such messages for Traces T1, T2, T3, and T4.

Table 3 shows the duration of a data transport phase for a video stream, the payload size per stream, and the average rate for the duration of a transport session. Session duration information is shown in the first row of Table 3 and Fig. 8a. The CDF data shows that for all traces, a large number of the sessions have a short duration (80% of the sessions shorter than 160 s). The mean session duration obtained in our study is significantly shorter than a mean session duration of 18.7 min reported by Gill et al. [6]. This is due to the fact that we only look at the duration for a single video stream while Gill et al. measure the time a user spends at the YouTube web page. The mean distribution observed in [6] could include the case where a viewer watches multiple video clips consecutively. The row *Payload Size* shows the amount of payload transported per session, including possible retransmissions due to packet losses. On average, the payload is on the order of several MB (between 6.3 and 7.5 MB). Distribution information about the payload size is shown in more detail in Fig. 8b. The last row in Table 3 shows that data was streamed with an average bandwidth between 630 Kbps and slightly larger than 900 Kbps. The CDF of the data rate (Fig. 8c) shows an interesting bimodal download rate. The CDF is separated into two regions. In the first region, videos are streamed with a rate of approximately 800 Kbps while, in the second region, videos are streamed with a rate of 1200 Kbps or higher. To further investigate the reason for this bimodal behavior, we conducted a fully-controlled measurement from a client computer connected to one of the wired networks in the student dorms. Since we focus on a single client only, we ran *tcpdump* directly on that client to obtain as much detailed information about the communication between the client and YouTube servers as possible. We chose the dorm location since roughly 2/3 of the requests for YouTube video clips originate from the dorm subnetworks. In this case, we

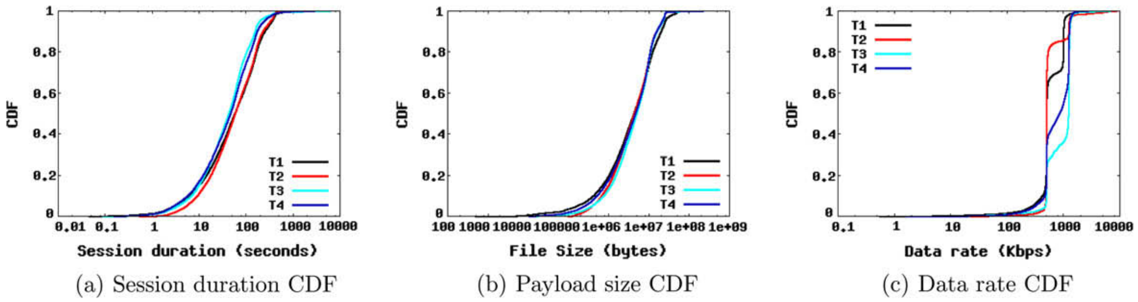


(a) Request per client (Trace N2)
(b) Request per client (Trace N4)
(c) Request per client (Trace N5)
(d) Request per client (Trace N6)

Fig. 7. Distribution of requests per client for 24-h measurement interval.

Table 3
Results from video stream flow analysis.

Trace		T1	T2	T3	T4
Duration (s)	Avg	99.62	95.81	81.34	75.12
	Max	4421.00	2359.83	16956.28	2834.19
	Min	0.04	0.53	0.04	0.15
Payload size (bytes)	Avg	7.5×10^6	6.4×10^6	6.3×10^6	6.5×10^6
	Max	2.15×10^8	1.30×10^8	1.42×10^8	2.18×10^8
	Min	484	95760	452	10304
Rate (Kbps)	Avg	632	646	908	841
	Max	5450	8633	10582	2696
	Min	0.54	6.74	0.19	1.6



(a) Session duration CDF
(b) Payload size CDF
(c) Data rate CDF

Fig. 8. CDF of the session duration (a), data rate (b), and file size (c) for traces T1, T2, T3, and T4.

again obtained a bimodal distribution for the data rate. The measured data rate is either around 700 Kbps or 1200 Kbps. For all the lower bandwidth cases, the video clips were transmitted from a server in the *googlevideo.com* domain while the higher bandwidth clips were served from servers in the *youtube.com* domain. From this result we conjecture that videos from servers in the *googlevideo.com* domain are transmitted with a lower rate limitation than those transmitted from the *youtube.com* domain.

4. Request modeling

Having presented and analyzed our YouTube measurement results, we focus in this section on modeling of YouTube traffic in an access network. The motivation for the modeling is to overcome the limitations of trace-driven simulations. For example, in order to investigate the scalability of the distribution schemes presented in Section 5, traces with a higher request rate or a larger user population would be necessary. It is difficult to obtain a wide range of traces in terms of request rates or the user population size from passive measurement data; synthetic traces are thus an attractive alternative. In the following sections, we consider how to generate such synthetic traces based on the statistical information we presented in Section 3. The analysis of these traces and the performance of the different distribution infrastructures is then presented in Section 5. Our model generates user requests that contain the following four pieces of information: video ID, client IP, request time, and content size. In this section, we define how each of these four components of a request is determined.

4.1. Video ID

Video ID is the unique identifier associated with each YouTube video requested by a client. Two-steps are required in order to obtain this information. The first step decides if the request is for a new video or for a previously requested video. In the second step, if the request is for a video previously requested it must be determined which video in the set of videos with multiple requests should be chosen. In the first step, information shown in the last two rows of Tables 1 and 2 is taken into account. For example, if one wants to create a new synthetic trace based on the information from trace N2, then 80% of the generated requests would be for new videos. To determine which of the videos requested multiple times should be chosen, the actual video popularity information is taken into account. Let us assume a new synthetic trace should be generated which creates twice the load of N2 (which would result in 27022 requests instead of 13511 in the original trace). In this case new video IDs for requests would be created as follows: 80% of the request would be for new videos (and a new unique ID is created). For the remaining 20% of the requests existing video IDs are chosen based on the popularity distribution from Section 3.2.

4.2. Clients

The IP address of the requesting client must also be determined in the synthetic trace. This IP address would be needed, for example, when simulating a P2P-based distribution architecture, in which videos are cached on client nodes and thus the exact node that caches the video must be specified. The determination of synthetic IP addresses is done in a manner similar to the determination of video IDs (see Section 4.1). First, information from the real measurement traces is used to determine how many requests are one-time requests and how many are repeated requests from a unique IP address. Based on this information it is decided if a new synthetic request is from a new IP address or from an address that has previously generated YouTube requests. For clients that make more than one request, the popularity distribution obtained from the measured traces is used so that new requests will be created with new IP addresses proportional to their popularity obtained from the trace data.

4.3. Request arrival time

We next describe a method to generate client requests at a higher rate. Synthetic traces with request rates higher than the ones in the real measurement traces are required to investigate the scalability of alternative distribution architectures. To determine the arrival times for the synthetic requests we used the following method. For a synthetic trace, we generate a new request between two consecutive requests in the original trace according to $t_{new}(i) = (t_{orig}(i) + t_{orig}(i+1))/2$, where t_{orig} represents the arrival times from the real trace data and t_{new} indicates the new arrival times for the synthetic trace. Fig. 6 shows the CDFs of request inter-arrival times for new synthetic traces S2, S4, S5, and S6. The request times for these synthetic traces were generated as described above. As shown in Fig. 6, this method roughly cuts the inter-arrival time in half. By relying on this method, periods of higher and lower request activity in the original data will also be reflected in the synthetic data.

4.4. Content size

In order to determine the file size of each requested video clip, we make use of the statistical information obtained from the traffic analysis presented in Section 3.6. More specifically, each time a new synthetic request is created, the size of the requested video clip is determined by randomly sampling the payload size CDF shown in Fig. 8b.

4.5. Synthetic traces

In this section, we give a brief overview on six traces that were created in a synthetic way as described above, based on the traces presented in Section 3 with twice the request rates observed in the real traces.[7] Performing simulations with such synthetic traces allows us to investigate the scalability of alternative distribution infrastructures. Assuming that the request rate for YouTube video clips from a campus network will actually double in the future, the results of these simulation can indicate the appropriateness of these distribution infrastructures. Table 4 gives an overview of the synthetically generated traces. The number of requests are twice as high when compared to the original traces (see Table 2). For example, Trace S2 contains 27022 requests while Trace N2 contains 13511 requests.

[7] Nevertheless, synthetic traces can be created with a wide variety of characteristics.

Table 4
Modeled YouTube traffic traces.

Trace		S1	S2	S3	S4	S5	S6
Length (h)		12	24	24	24	24	24
# Of unique clients		1553	2520	2734	7051	16184	19776
Per video stats	Total	25976	20244	8954	27045	57102	74514
	Single (%)	78.1	79.6	81.1	79.0	71.8	71.6
	Multi (%)	21.9	20.4	18.9	21.0	28.2	28.4
Request stats	Total	36080	27022	11542	37500	92814	122280
	Single (%)	56.2	59.6	62.8	56.9	44.2	43.6
	Multi (%)	43.8	40.4	37.2	43.1	55.8	56.4

5. Distribution infrastructures

In this section, we use the synthetically generated traces to study the performance of distribution infrastructures (e.g., local caching schemes) that are different from the one used by the actual YouTube system. Two different classes of data sets are used for this evaluation. The first class is composed of the 24-h snapshot traces (N2, N4, N5, and N6). The second one consists of a set of synthetic traces created using the method described in Section 4. Based on the two classes of traces, in the following sections, results from trace-driven simulations for video distribution based on peer-to-peer and proxy caching are presented.

5.1. Peer-to-peer (P2P)-based caching

In P2P-based caching, a video clip that is requested by a client can be served from another client if the clip is kept in the latter client's storage. For example, if client 2 requests a video clip that has also been downloaded by client 1 earlier, client 2 can retrieve that video directly from client 1's cache rather than a YouTube content server. In [1], we considered a very simple P2P caching approach. In this simple approach, a client only caches a requested video if it has not been cached by another client in the campus network. Thus, no more than one copy of a video is cached in the campus network. To investigate if the P2P caching approach could be further improved, we consider an improved caching scheme in which a client always caches a requested video if it cannot be served from another client in the same access network (in our case the campus network). Note that a video can be cached at another client but might not be available due to the client being off-line. This may result in multiple copies of one video being cached in multiple clients, which increases the probability of serving a new request for the video from such clients. This effect is reflected in the results we obtained via simulation of this improved P2P approach as shown in Fig. 9. A comparison with the results from the simple P2P approach shows that the hit rate significantly improves. Due to space restrictions we omit the detailed explanation of how the client availability is determined and refer the interested reader to [1].

5.2. Proxy caching

Under proxy caching, client requests for YouTube content from within the campus network would be redirected to the proxy. Each request is analyzed at the proxy cache and the following actions are taken. In the case that the requested video clip has already been stored on the proxy, it is directly transmitted from the proxy to the client. If the requested video clip has not been cached, the proxy can make two decisions. Based on its local caching strategy the proxy may decide to cache the requested video clip. In this case, it forwards its own request for the video clip (originally requested by the client) to the YouTube web server. Upon reception of the data stream, the proxy starts storing the video payload on its local storage and also forwards the data to the client. If the proxy decides not to cache the requested video clip, it does not intervene in the message exchange between the client and the YouTube content server.

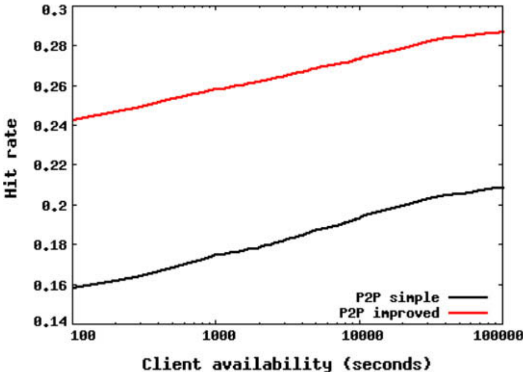


Fig. 9. Cache hits for improved P2P caching approach. The *x*-axis shows different intervals for peer availability. Simulation is based on data from T3.

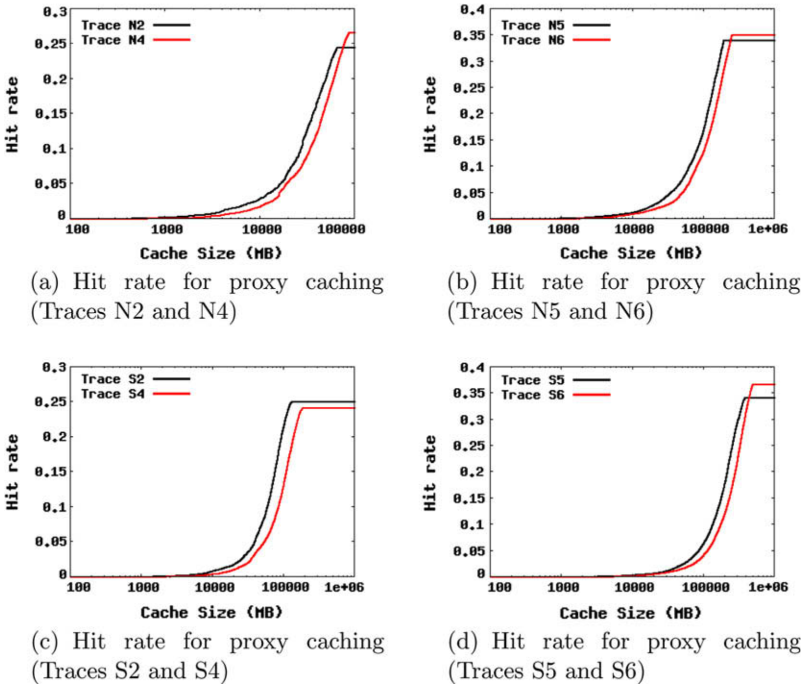
In contrast to the proxy-based simulations we presented in [1], we refine the simulation in terms of the file size for the cached video clip. In [1], we assumed that all files have the same size (the average file size shown in Table 3). In our simulations here the file size for each video clip is determined by the method described in Section 4.4. Since we could not obtain information about the stream traffic available from Traces T5 and T6 (see Section 3.6), we use the statistical information obtained from T2 for N2 and S2, and from T4 for N4, N5, N6 and S4, S5, S6. We are aware that while this might not necessarily reflect the exact scenario for Traces N4, N5, N6, S4, S5, and S6 in terms of file size, we do not expect a significant change in the file size distribution over time.

We conducted a simulation of the proxy cache-based distribution architecture for YouTube video clips where the total storage size of the cache located at the gateway grows from 100 MByte to 1 Tbyte. The results of this simulation are shown in Fig. 10. The *x*-axis of the four graphs shows the storage space while the hit rate is shown on the *y*-axis. For cache replacement a simple FIFO strategy is employed. Fig. 10a shows the simulation results for Trace N2 and N4. As expected the hit rate constantly increases with an increasing cache size. A comparison of the results for N2 and N4 shows that a larger cache is required for N4 due to the higher rate of unique overall requests. On the other hand the results show the fact that N4 has higher rate of video clips that get requested multiple times (last row of Table 2). Consequently, this higher request rate leads to a higher final hit rate (26.5%). Compared to the traces with high request rates (N5 and N6 in Fig. 10b) a maximum cache size of less than 1 GByte is required to achieve the maximum hit rate. The comparison between the original traces and the synthetic traces (e.g., Fig. 10a and c) shows that the caching behavior is almost identical. This is as expected since the statistics for original and synthetic traces are almost identical. Nevertheless, the hit rates are not exactly identical due to some slight differences between original and synthetic traces. The results of the synthetic traces also show how proxy caching behaves in case of an increased request rate for video clips in the campus network.

6. Related work

A major research area that is related to our work is concerned with the analysis and characterization of streaming services in the Internet. Early work in this area is focused on the characterization of videos on the web [7] and how users access such videos [8]. Chesire et al. [9] apply a method very similar to our approach to measure streaming media workload in their university network. In their study, RTSP messages are monitored to collect traces. These traces are then analyzed characterizing the streaming media workload. In addition, the trace data is used as an input to simulate a caching system to study the benefits of such an approach. An analysis of RealVideo performance with traces of clients from different geographical locations is presented by Wang et al. [10], which use an active measurement method. Users were asked to use a modified version of the RealPlayer that monitors the requested video stream. Results from this investigation give hints about transport characteristics (available bandwidth and resulting quality of video) of the connection between the video servers and clients. Another active measurement approach is presented by Li et al. [11], where a media crawler is used to retrieve audio and video clips from the web. Our approach is different since we monitor requests from *all* users in a university network to obtain information such as the local popularity of video clips. In addition, we collect low level information about the individual streaming sessions at the transport layer.

In addition to our work, three YouTube measurement studies have recently been reported in literature. The approach of Cha et al. [2] differs from ours since only information directly available from YouTube servers is used to analyze the characteristics of videos served by the YouTube servers. Due to their focus on the global nature of YouTube traffic, their work does not shed light on the



(a) Hit rate for proxy caching (Traces N2 and N4)

(b) Hit rate for proxy caching (Traces N5 and N6)

(c) Hit rate for proxy caching (Traces S2 and S4)

(d) Hit rate for proxy caching (Traces S5 and S6)

Fig. 10. Hit rate based on proxy cache size.

possible benefit of using caching mechanisms in the access networks where YouTube clients are physically co-located. The motivation for the Gill et al. [3] measurement study is quite similar to ours. However, unlike ours, their trace contains a limited view of their gateway traffic from YouTube servers since only a predefined set of YouTube servers could be monitored. A similar general trend between their and our results is observable. Due to the different measurement mechanisms used in both studies, it is hard to make a detailed comparison of the results. In contrast to our work, neither of [2,6] quantitatively measures the benefits of different distribution architectures based on simulations. Gill et al. also conducted a study that investigates the session characteristics of YouTube users [6]. In their study the focus is on user sessions, i.e., a request issued by a user to a web site in a single visit to the site. Comparisons with previous results on user sessions show that users take longer to decide which page to visit next after watching a video clip and a change in workload characteristics. Finally, we recently published our measurement results and trace-based simulations in [1]. In contrast to this article, a series of measurements that were performed over the series of two month were presented. Thus, any long-term trends in YouTube traffic characteristics could not be observed. In addition, simulations were only performed based on real measurement traces. In this article, we also make use of synthetically generated traces to study the performance of distribution techniques on possible future traffic loads.

The generation of synthetic traces to mimic the characteristics of real traces is not new and has been used before. Some of the work in this area has focused on the synthetic generation of web traces [12–14], while other work is more general and focuses on synthetic network traffic traces [15]. The work presented in this article applies the same ideas as used in the works above but focuses on the generation of traces for a single web application.

7. Conclusion

In this article, we have characterized the nature of YouTube traffic in a large university campus network. We have described a methodology to monitor YouTube signaling and data traffic between clients in our campus network and YouTube servers. Measurements were performed over a 10-month period. Based on these measurements, we analyzed the duration and the data rate of streaming sessions, the popularity of videos, and access patterns for video clips from the clients in the campus network. Results from this analysis show that trace statistics are relatively stable over a short-term observation period while long-term trends can be observed for a larger observation period. Our long-term observation complies with other reports on the increase of the popularity of the YouTube service. In addition, we demonstrate how the statistical information gained by the analysis of the trace data can be used to create synthetic traces. Performing trace-based simulations, we investigate the performance of alternative local caching methods for video traffic. In the case of proxy caching we do not only use real traces but also synthetic traces to demonstrate how this caching method would perform with future YouTube traffic loads.

The results of our study show that (i) local and global popularity are not or only slightly correlated, (ii) popularity of the YouTube service in a campus network is still increasing, (iii) alternative distribution methods can reduce traffic between the campus network and the Internet significantly.

Acknowledgements

This work was supported in part by the National Science Foundation under ANI-0240487, ANI-0325868, CNS-323597, and CNS-0626874. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation. The authors would also like to thank the anymous reviewers for their valuable comments.

References

[1] M. Zink, K. Suh, Y. Gu, J. Kurose, Watch global, cache local: YouTube network traffic at a campus network – measurements and implications, in: Proceedings of SPIE/ACM Conference on Multimedia Computing and Networking (MMCN), Santa Clara, USA, 2008.

[2] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, S. Moon, I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system, in: Proceedings of ACM Internet measurement Conference(IMC), San Diego, CA, USA, 2007.

[3] P. Gill, M. Arlitt, Z. Li, A. Mahanti, YouTube traffic characterization: a view from the edge, in: Proceedings of ACM Internet measurement Conference(IMC), San Diego, CA, USA, 2007.

[4] M. Zink, K. Suh, Y. Gu, J. Kurose, Watch global, cache local: YouTube network traffic at a campus network: measurements and implications, in: Technical Report 07-39, Department of Computer Science, University of Massachusetts Amherst, 2007.

[5] Endace DAG Network Monitoring Interface. URL <http://www.endace.com>.

[6] P. Gill, Z. Li, M. Arlitt, A. Mahanti, Characterizing users sessions on YouTube, in: Proceedings of SPIE/ACM Conference on Multimedia Computing and Networking (MMCN), Santa Clara, USA, 2008.

[7] S. Acharya, B. Smith, Experiment to characterize videos stored on the web, in: Proceedings of SPIE/ACM MMCN, San Jose, CA, USA, 1998, pp. 166–178.

[8] S. Acharya, B. Smith, P. Parnes, Characterizing user access to videos on the world wide web, in: Proceedings of SPIE/ACM MMCN, San Jose, CA, USA, SPIE, 2000, pp. 130–141.

[9] M. Chesire, A. Wolman, G. Voelker, H. Levy, Measurement and analysis of a streaming-media workload, in: Proceedings of USENIX Symposium on Internet Technologies and Systems, San Francisco, CA, USA, 2001.

[10] Y. Wang, M. Claypool, Z. Zhu, An empirical study of realvideo performance across the internet, in: Proceedings of ACM Internet Measurement Workshop, San Francisco, CA, USA, 2001, pp. 295–309.

[11] M. Li, M. Claypool, R. Kinicki, J. Nichols, Characteristics of streaming media stored on the web, ACM Transactions on Internet Technology 5 (4) (2005) 601–626.

[12] L. Cherkasova, G. Ciardo, Characterizing temporal locality and its impact on web server performance, in: Proceedings of the IEEE ICCCN 2000 Conference, Las Vegas, NV, USA, 2000.

[13] W. Liu, C.T. Chou, Z. Yang, X. Du, Popularity-wise proxy caching for interactive media streaming, in: Proceedings of the LCN 2007 Conference, Tampa, FL, USA, 2004.

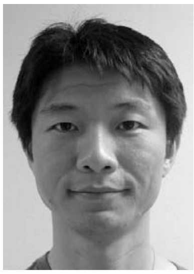
[14] S. Weber, R. Hariharan, A new synthetic web server trace generation methodology, in: Proceedings of the ISPASS 2003 Conference, Austin, TX, USA, 2003.

[15] K.V. Vishwanath, A. Vahdat, Realistic and responsive network traffic generation, in: Proceedings of the SIGCOMM 2006 Conference, Pisa, Italy, 2006.

Michael Zink is currently a research assistant professor in the Computer Science Department at the University of Massachusetts in Amherst. Before, he worked as a researcher at the Multimedia Communications Lab at Darmstadt University of Technology. He works in the fields of sensor networks and distribution networks for high bandwidth data. Further research interests are in wide-area multimedia distribution for wired and wireless environments and network protocols. He is one of the developers of the KOMSSYS streaming platform. Michael Zink received his Diploma (M.Sc.) from Darmstadt University of Technology in 1997. From 1997 to 1998 he was employed as guest researcher at the National Institute of Standards and Technology (NIST) in Gaithersburg, MD, where he developed an MPLS testbed. In 2003, he received his Ph.D. degree (Dr.-Ing.) from Darmstadt University of Technology, his thesis was on "Scalable Internet Video-on-Demand Systems".

In April 2004 he joined the CASA ERC at the University of Massachusetts as Technical Integration Leader for the first DCAS radar network becoming operational in spring 2006 in central Oklahoma.

Kyoungwon Suh received B.S. and M.S. degrees in Computer Engineering from Seoul National University in Korea in 1991 and 1993, respectively. He continued his studies in the Department of Computer Science at Rutgers University in NJ, where he earned a M.S. degree in 2000. In 2007, he received his Ph.D. degree in Computer Science from University of Massachusetts at Amherst. He also has served as a technical consultant to NHN Corporation in the area of network and system security in 2008. He is currently an assistant professor in Illinois State University, Normal, IL. His present research interests include peer-to-peer and overlay networks, network measurement and inference, network security, and multimedia content distribution. He is a member of ACM and IEEE. http://www.itk.ilstu.edu/faculty/kwsuh.



Yu Gu received B.S. and M.S. degrees in Computer Science from Beijing University of Aeronautics and Astronautics in 1998 and 2001, respectively. In 2008, he received his Ph.D. degree in Computer Science from the computer network research group in the University of Massachusetts, Amherst. He is now with NEC Labs America. His research interests include network measurement, congestion control, anomaly detection, network simulation and multimedia networks.

Jim Kurose is currently Distinguished University Professor (and past chairman) in the Department of Computer Science at the University of Massachusetts, where he is also Associate Director of the NSF Engineering Research Center for Collaborative Adaptive Sensing of the Atmosphere (CASA). Professor Kurose has been a Visiting Scientist at IBM Research, INRIA, Institut EURECOM, the University of Paris, LIP6, and Thomson Research Labs. His research interests include network protocols and architecture, network measurement, sensor networks, multimedia communication, and modeling and performance evaluation. He has served as Editor-in-Chief of the IEEE Transactions on Communications and was the founding Editor-in-Chief of the IEEE/ACM Transactions on Networking. He has been active in the program committees for IEEE Infocom, ACM SIGCOMM, and ACM SIGMETRICS conferences for a number of years, and has served as Technical Program Co-Chair for these conferences. He has received a number of awards for his educational activities, including the IEEE Taylor Booth Education Medal. He is a Fellow of the IEEE and the ACM. With Keith Ross, he is the co-author of the textbook, "Computer Networking, a top down approach" published by Addison-Wesley Longman.