Characterizing traffic flows originating from large-scale video sharing services

Tatsuya Mori[1] and Ryoichi Kawahara[1], Haruhisa Hasegawa[1], Shinsuke Shimogawa[1]

NTT Research Laboratories, 3–9–11 Midoricho, Musashino-city, Tokyo 180–8585, JAPAN,
{mori.tatsuya, kawahara.ryoichi, hasegawa.haruhisa,
shimogawa.shinsuke}@lab.ntt.co.jp

Abstract. This work attempts to characterize network traffic flows originating from large-scale video sharing services such as YouTube. The key technical contributions of this paper are twofold. We first present a simple and effective methodology that identifies traffic flows originating from video hosting servers. The key idea behind our approach is to leverage the addressing/naming conventions used in large-scale server farms. Next, using the identified video flows, we investigate the characteristics of network traffic flows of video sharing services from a network service provider view. Our study reveals the intrinsic characteristics of the flow size distributions of video sharing services. The origin of the intrinsic characteristics is rooted on the differentiated service provided for free and premium membership of the video sharing services. We also investigate temporal characteristics of video traffic flows.

1 Introduction

Recent growth in large-scale video sharing services such as YouTube [19] has been tremendously significant. These services are estimated to facilitate hundreds of thousands of newly uploaded videos per day and support hundreds of millions of video views per day. The great popularity of these video sharing services has even lead to a drastic shift in Internet traffic mix. Ref. [5] reported that the share of P2P traffic dropped to 51% at the end of 2007, down from 60% the year before, and that the decline in this traffic share is due primarily to an increase in traffic from web-based video sharing services. We envision that this trend will potentially keep growing; thus, managing the high demand for video services will continue to be a challenging task for both content providers and ISPs.

On the basis of these observations, this work attempts to characterize the network traffic flows, originating from large-scale video sharing services as the first step toward building a new data-centric network that is suitable for delivering numerous varieties of video services. We target currently prominent video sharing services; YouTube in US, Smiley videos in Japan [16], Megavideo in Hong kong [12], and Dailymotion in France [6]. Our analysis is oriented from the perspective of a network service provider, i.e., we aim to characterize the traffic flows from the viewpoints of resident ISPs or other networks that are located at the edges of the global Internet.

Our first contribution is identifying traffic flows that originate from several video sharing services. The advantage of our approach lies in its simplicity. It uses source

IP addresses as the key for identification. To compile a list of IP addresses associated with video sharing services, we analyze a huge amount of access logs, collected at several web cache servers. The key idea behind our approach is to leverage the naming/addressing conventions used by large-scale server farms. In many cases, web servers for hosting videos and those for other functions such as managing text, images, or applications, are isolated. These servers are assigned different sets of IP prefixes, which are often associated with intrinsic hostnames, e.g., “img09.example.com” is likely used for servers that serve image files. We also leverage open recursive DNS servers to associate the extracted hostnames of video hosting servers with their globally distributed IP addresses.

Our second contribution is revealing the intrinsic characteristics of video sharing services, which are not covered by conventional web traffic models. The origin of the characteristics is based on the differentiated services provided for free and premium membership of the video sharing services. We also investigate temporal characteristics of video traffic flows.

The remainder of this paper is structured as follows. Section 2 describes the measurement data set we used in this study. We present our classification techniques in section 3. We then analyze the workload of video sharing services, using the identified traffic flows originating from video hosting servers, in section 4. Section 5 presents related work. Finally, section 6 concludes this paper.

2 Data description

This section describes the two data sets we used in this study. The first data set was web proxy logs, which enable us to collect the IP addresses of video hosting servers used by video sharing services. The second data set was network traffic flows, which enable us to investigate the characteristics of the workload of video sharing services.

2.1 Web cache server logs

We used IRCache data set [10], which is web cache server logs, open to the research community. We used the access logs collected from 7 root cache servers located in cities throughout the US. The access logs were collected in September 2009. Since the client IP addresses were anonymized for privacy protection, and the randomization seeds are different among files, we could not count the cumulative number of unique client IP addresses that used the cache servers. We noted, however, that a typical one-day log file for a location consisted of 100–200 unique client IP addresses, which include both actual clients and peered web cache servers deployed by other institutes. Assuming there were no overlaps of client IP addresses among the web cache servers, the total number of unique client IP addresses seen on September 1, 2009 was 805, which was large enough to merit statistical analysis.

The one-month web cache logs consisted of 118 M web transactions in total. The 118 M transactions account for 7.8 TB of traffic volume. 89 M transactions return the HTTP status code of “200 OK” and these successfully completed transactions account for 6.2 Terabytes of traffic flows that are processed on the web cache servers.

2.2 Network flow data

In this work, we define a flow as a unique combination of source/destination IP address, source/destination port number, and protocol. We used network flow data that were collected at an incoming 10-Gbps link of a production network. For each flow, its length in seconds and size in bytes were recorded. The measurement was conducted for 9.5 hours on a weekday in the first quarter of 2009. The format of the network flow data set is: {`ctime, mtime, src IP, anonymized client ID (AID), protocol, src port, dst port, #pkts, bytes`}, where “`ctime`” and “`mtime`” are created and modified time of a flow, and “`#pkts`” and “`bytes`” are the number of packets and bytes of a flow, respectively. “`AID`” is randomized destination (client) IP address. The 5-tuple, {`src IP, AID, protocol, src port, dst port`} composes a flow.

The total amount of incoming traffic carried during the measurement period was 4.4 TB, which corresponded to the mean offered traffic rate of 1.03 Gbps. The incoming traffic consisted of 108 M distinct flows that were originated from 5.5 M of sender IP addresses to 34 K of receiver (client) IP addresses. Of these, 40.6 M were the incoming web flows. The traffic volume of the web flows was 1.8 TB (mean offered traffic rate was 0.42 Gbps).

3 Extracting sources of video flows

We now present the techniques for identifying video flows among the network flow data set. We use a source IP address as a key for identification, i.e., if an incoming flow originates from an IP address associated with a video sharing service, we identify the flow as a video flow.

As mentioned earlier, the key idea of this approach is leveraging the naming/addressing conventions used by large-scale web farms, where servers are grouped by their roles, e.g., hosting a massive amount of large video files, hosting a massive number of thumbnails, or providing rich web interfaces. We first present a video sharing service uses distinct hostnames for each subtypes of HTTP content-type, i.e., video, text, image, and application. We then collect hostnames of video hosting servers. Finally, we compile the IP addresses that are associated with the hostnames.

3.1 Classifying hostnames with subtypes of HTTP content-type

This section studies naming convention used in large-scale video sharing services and presents distinct hostnames are used for each sub-category of HTTP content-type. The web cache server logs are used for the analysis. We also study the basic property of the objects for each category.

We start by looking for a primary domain name for the video sharing service of interest, e.g., YouTube. More specifically, we compare a hostname recorded in web cache logs with that domain name to see if the hostname in URL matches the regular expression in perl-derivative, `/\.youtube\.com$/`. If we see a match, we regard the object as one associated with YouTube. Although we use the YouTube domain as an example in the following, other prominent video sharing services today can also be explored in a similar way. For brevity, only the results for those services will be shown later.

Table 1. Statistics of Content-types in web transactions associated with YouTube.

Content-type	No. of transactions	total volume	mean size
video	160,180	681 GB	4.3 MB
image	48,756	157 MB	3.2 KB
text	458,357	4.3 GB	9.5 KB
application	109,743	359 MB	3.3 KB
other	35,021	23 MB	678 B

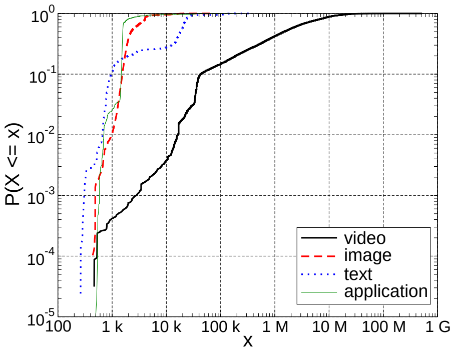

Fig. 1. CDFs of object size for each content-type.

We analyzed the 89 M of successfully processed HTTP transactions, and found 812 K transactions were associated with YouTube. The total volume of these transactions was 686 GB in the one-month logs. Frequency of content-types for the transactions to the YouTube domain are summarized in Table 1. A content-type is defined as an Internet media type, which is used by several protocols such as SMTP, HTTP, RTP, and SIP. Examples of observed sub-types for each content-type are x-flv/mp4 (video), jpeg/gif (image), html/xml (text), and x-shockwave-flash/javascript (application). This variety of content files together forms the video sharing service. Notice that more than 80% of YouTube web transactions carry non-video data. This indicates that these non-video data are crucial factors from the viewpoint of processing overhead rather than transport overhead. For instance, from the perspective of a network service provider, these non-video transactions consume a lot of resources on network middle boxes, such as firewalls or NAT, which need to keep track of connections.

Next, we looked at the correlation between the content-type and size of an object. In addition to the total volume and mean size of objects, we plot the cumulative distribution functions (CDFs) of object sizes for each content-type (see Fig. 1). Notice that the majority of image, text, and application objects are small. For instance, 99% of image and application objects are less than 10 KB, and 99% of text objects are less than 30 KB. In contrast, the size of video objects is heavy-tailed, ranging over 6 orders of magnitude, from less than 1 KB to 500 MB. We note that very small video objects, e.g., less than 20 KB, are likely due to partial or incomplete transmission.

Table 2. Number of distinct hostnames observed for YouTube transactions with HTTP status code of 200 or 206.

Content-type	# of hostnames
video	490
image	63
text	23
application	5

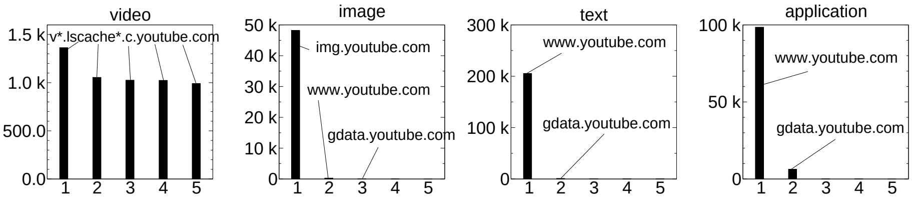

Fig. 2. Top 5 hostnames by number in YouTube web transactions for each content-type.

3.2 Collecting hostnames of server farm

Using the naming convention studied in the previous subsection, we aim to compile a list of hostnames of video hosting servers from the web cache server logs. Processing the data set is straightforward, however, we note that it is necessary to cope with side effects of irregular patterns such as HTTP status code “204/No Content”, which is used in cases where the request was successfully processed but the response does not have a message body [10]. To avoid this, we prune the transactions that have HTTP status codes other than “200/Ok” or “206/Partial Content”, which mean the successful transaction and the response to a request to an object data subset, respectively. This heuristic eliminates the cases where video hosting servers return text/html content with the “204/No Content” code, “303/redirection” code, or other error codes.

The statistics of collected hostnames for YouTube are shown in Table 2. While video objects are served by a large number of servers, objects of other content-types are served by a small set of servers. Next, we focus on the number of web transactions per hostname. The top five hostnames for each content-type are shown in Fig. 2. Clearly, the hostnames of the video hosting servers are different from those for other content-types. We also notice that for video hosting servers, the number of accesses are balanced among the top five servers; this indicates that the video hosting servers are likely to be accessed by load-balancing mechanisms.

We next extract the naming rule of the video hosting servers from the collected hostnames. Table 3 shows the hostnames of the video hosting servers of YouTube, where “⊙” and “⊗” represent the variables of a number. In compiling the list, we complement the missing numbers in hostnames. For instance, if we observe “`foo1.example.com`” and “`foo3.example.com`”, but not “`foo2.example.com`”, we conjecture that the last hostname was likely missed during data measurement and add it to the list. We also test whether the complemented hostname has a valid DNS A record(s). In total, the generalized hostnames of YouTube video hosting servers contribute 998 distinct hostnames. We note the primary classes of hostnames are significantly biased to the top two classes, i.e., “v⊙.lscache⊗.c” and “v⊙.cache⊗.c”. The number of hostnames for these classes is

Table 3. Generalized hostnames of YouTube video hosting servers and number of observed transactions for each class of hostname.

Hostnames	Complemented range	# of observed transactions
v⊙.lscache⊗.c	$1 \le \odot \le 24$, $1 \le \otimes \le 8$	130,286
v⊙.cache⊗.c	$1 \le \odot \le 8$, $1 \le \otimes \le 8$	27,485
tc.v⊙.cache⊗.c	$1 \le \odot \le 24$, $1 \le \otimes \le 8$	1626
v⊙.nonxt⊗.c	$1 \le \odot \le 24$, $1 \le \otimes \le 8$	25
lax-v⊙.lax	$1 \le \odot \le 308$	19
sjl-v⊙.sjl	$1 \le \odot \le 50$ (with exceptions)	19

Table 4. Generalized hostnames of video hosting servers.

Service	Hostnames	Complemented range
Smiley videos	smile-com⊙⊗.nicovideo.jp	$0 \le \odot \le 6, 0 \le \otimes \le 3$
Smiley videos	smile-cub⊙⊗.nicovideo.jp	$0 \le \odot \le 6, 0 \le \otimes \le 3$
Megavideo	www⊙.megavideo.com	⊙ can be any positive integer.
Dailymotion	proxy-⊙⊗.dailymotion.com	$0 \le \odot \le 9, 0 \le \otimes \le 9$

$256\ (= (24+8) \times 8)$. In the rest of this work, we will use these 256 names as the primary hostnames of YouTube.

Using these techniques, we extracted generalized hostnames of video hosting servers for other video sharing services. The results are summarized in Table 4. Although naming conventions differ among the services, all the services use their own naming rule. We finally note that these lists are snapshots and should be periodically updated.

3.3 Extracting global IP addresses

This section aims to extract global IP addresses that are associated with the hostnames collected in the previous subsection. It is well known that large-scale server farms such as YouTube and Akamai typically use a large number of global IP addresses that are associated with a smaller set of hostnames. The methodology is aimed at balancing the load of globally distributed accesses across the server farms [9]. This addressing convention enables us to associate global IP addresses with particular hostnames that are used for video hosting.

Because of these spatially distributed load-balancing mechanisms used in server farms, an IP address obtained by looking up the DNS A record of a hostname could differ in location. For example, Akamai CDN tweaks the DNS mechanism to select the closest web server from a client. Similarly, YouTube uses the HTTP redirection mechanism to introduce load balancing in a dynamic way [20]. Thus, we need to perform globally distributed DNS resolutions to compile a list of IP addresses associated with the list of hostnames.

To achieve this objective, we adopted a methodology proposed by Huang et al. in Ref. [9]. They performed globally distributed DNS resolution of 16 M unique web hostnames to obtain a complete list of DNS CNAMEs for Akamai CDN servers, which could be used to estimate roughly the scale of the Akamai infrastructure. The key idea of their approach is to leverage open recursive DNS (ORDNS) servers, which will resolve DNS queries for any clients from anywhere; this approach enables us to obtain global

Table 5. Number of IP addresses for video hosting servers.

Service	# of addresses
YouTube	2,138
Smiley videos	74
Megavideo	670
Dailymotion	100

view of the system from an Internet edge site. We use a similar approach to that shown in Ref. [9] to compile the list of ORDNS servers. In total, we collected more than 5,000 ORDNS servers that are distributed across 68 countries.

For each hostname shown in Tables 3 and 4, we performed DNS resolutions from all the ORDNS servers we collected, and compiled the resolved answers. The results are summarized in Table 5. Notice that these services consist of a fairly large number of servers. For example, YouTube has 2,138 unique IP addresses, which is much larger than the original 256 hostnames. Note that the number of global IP addresses does not necessarily correspond to the number of actual video hosting servers, meaning the actual infrastructure could be larger than can be seen from an Internet edge site. In addition, the extracted IP addresses for video hosting servers are mostly different from those for other media types, i.e., image, text, and application. Thus, the obtained IP addresses of video hosting servers can be used as a simple and effective key to identify the video flows of large-scale video sharing services.

4 Characterizing video flows

In the previous section, we compiled a list of IP addresses for the video hosting servers, using web cache server logs. This section uses our network flow data set to characterize traffic flows originating from video sharing services by using the list of IP addresses. First, we study fundamental statistics of video flows, which plays an essential role in understanding the structure of traffic flows. Next, we employ in-depth analysis of flow size distributions, which exhibit intrinsic characteristics. Finally, we investigate temporal characteristics of video traffic flows.

4.1 Flow statistics

We investigate the fundamental statistics of video flows, i.e., flow size, flow rate, and flow duration, which form essential parts of the traffic workload model. Since a large portion of the extracted flows is composed of small flows, which could be incomplete flows or error flows, we exclude these small flows from our analysis. On the basis of the observation from Fig. 1, we use 20 KB as a threshold for pruning the small flows. As a result, of the 103K of collected YouTube video flows, 60K flows were removed. We note that although the number of pruned flows was not small, their contribution to the total traffic volume was negligible. Actually, the pruned flows contributed less than 1% in total traffic volume. We also note that majority of the pruned flows were *incomplete*, i.e., most of them were one-packet TCP flows with SYN/ACK flag, originated from youtube video hosting servers; i.e., the video hosting servers were likely port-scanned by some of clients. Since we are interested in the impact of YouTube traffic from the

Table 6. Statistics of observed flows that are larger than 20 KB.

Service	# of flows	Mean size	Mean rate	Mean duration
YouTube	43,960	4.1 MB	1.3 Mbps	41.8 sec
Smiley videos	3,438	21.3 MB	2.6 Mbps	139.8 sec
Megavideo	1,354	30.0 MB	1.3 Mbps	232.6 sec
Dailymotion	730	13.7 MB	1.5 Mbps	96.0 sec
All web	5,043,927	0.33 MB	0.9 Mbps	16.5 sec

Table 7. Uploading limitations (as of 1Q 2009).

Service	Free membership	Premium membership
YouTube	10 minutes or 2 GB per video	20 GB per video (partners)
Smiley videos	40 MB per video	100 MB per video
Megavideo	100 MB per video	5 GB per video
Dailymotion	20 minutes or 150 MB per video	–

network service provider perspective, focusing on flows that deliver actual video data is essential.

The basic statistics for these metrics are summarized in Table 6. In general, video flows are larger, faster, and longer than conventional web flows. This observation agrees with the previous work [15] by Plissonneau et al. Next, we look into the detailed characteristics of each metric.

Flow size: The top-left graph of Fig. 3 presents log-log complementary cumulative distribution function (CCDF) plots for flow size distributions. While the web flows (shown as "All web" in the legend of the graph) obey a clear Pareto-like distribution with moderate decaying at the tail, all the other video flows exhibit different characteristics. In general, they are significantly heavy-tailed; for instance, more than 60% of the video flows are larger than 1 MB for all video services, while less than 3% of flows are larger than 1 MB in all the web flows. This significant heavy-tailedness can also be seen in the flows of P2P file-sharing applications. What makes the video flows distinguishable from P2P flows is shown next. That is, the video flows exhibit clear change points in the middle area, where probability distributions drop sharply, i.e., the points at 30, 40, and 100 MB. In fact, we find that these change points correspond to the intrinsic capacity limitation of the video sharing services. This data is summarized in Table 7. We can see that the file size limitations agree with the change points of the flow size distributions. For example, the change point of Megavideo flow size distribution is 100 MB, which is exactly the upload file size limitation for non-premium (free) membership.

We conclude the following from these observations:

- Size distributions of video flows are quite heavy-tailed.
- The tail parts of flow size distributions for video sharing services are constrained by the limitation of available service resources for free membership (upload file size).

Flow rate: Next, we look at the flow rate, which is the number of bits divided by flow duration. The top-right graph of Fig. 3 shows log-log CCDF plots for flow

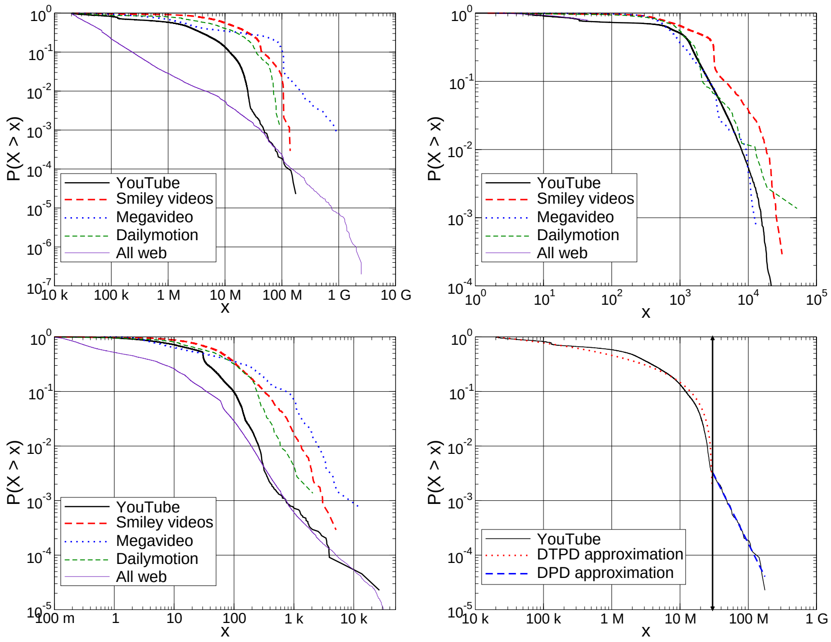

Fig. 3. Statistics of YouTube flows; Log-log CCDFs of flow size in bytes (top left), flow rate in Kbps (top right), and flow duration time in seconds (bottom left). Approximation of YouTube flow size distribution (bottom right).

rate distributions. In contrast to flow size distributions, we do not see much difference among the flows. An exception is a change point of 3 Mbps for Smiley videos. This observation again agrees with the differentiated service offered by the providers, i.e., premium members enjoy high-speed downloading while free members do not. We note that all the distributions fit the log-normal distributions well. For brevity, we omit the results.

Flow duration: Finally, we look into the flow duration. The bottom-left graph of Fig. 3 presents log-log CCDF plots of flow duration distributions. While we have seen the effects of available service capacities on the distributions for size and rate, we do not see the effects in the flow duration, despite the fact that the mean size and mean duration are positively correlated (see Table 6). Note that the actual download time (i.e., flow duration) may depend on other factors, such as throughput of access links or CPU resources of the end-terminal devices, which could be drastically different among the clients. Therefore, we do not see clear change points for the flow duration distributions.

4.2 Characterizing the size distributions of video flows

In the previous section, we found that the flow size distributions of current video sharing services have an intrinsic property. That is, the tail part of the distributions is constrained by the available service capacity, e.g., upload file size limitation. In this section, we attempt to characterize the flow size distribution to better understand its structure.

We start by approximating the distribution with known distributions. Assume that flows for free and premium membership can be modeled with different distribution models. Since flow sizes for free membership are truncated at a certain threshold, we adopt the discrete truncated Pareto distribution (DTPD) [13] for this class. For flow sizes for premium membership, we adopt the simple discrete Pareto distribution (DPD), which does not include any truncations.

Let X be a discrete random variable, which represents the size of a flow. The probability mass function of DTPD is given as

$$P(X = x) = f(x; \alpha_1, \beta_1, \theta_1) = \frac{x^{-\theta_1} - (x+1)^{-\theta_1}}{\alpha_1^{-\theta_1} - \beta_1^{-\theta_1}},$$

which satisfies $P(X > \alpha_1) = 1$ and $P(X > \beta_1) = 0$. Note that $P(X > x) = (x^{-\theta_1} - \beta^{-\theta_1})/(\alpha_1^{-\theta_1} - \beta_1^{-\theta_1})$ and $P(X = x) = P(X > x) - P(X > x + 1)$. The property of DTPD enables us to capture both the heavy-tailedness and the constraints at the threshold β_1. Similarly, the distribution function of DPD is given as

$$P(X = x) = g(x; \alpha_2, \theta_2) = \frac{x^{-\theta_2} - (x+1)^{-\theta_2}}{\alpha_2^{-\theta_2}}.$$

We now illustrate how the flow size distribution can be approximated with the DTPD and DPD in a graphical manner. YouTube is chosen as our case study.

We first set $\alpha_1 = 20,000$ and $\beta_1 = 30,000,000$ (bytes), which are the minimum flow size (20 KB) we are interested in, and the graphically estimated truncation point, respectively. Note that YouTube has two-way constraints: size and time; thus, the truncation point reflects their mix. In general, β_1 reflects the capacity limitation for a video sharing service.

Next, we estimate the parameters of DTPD and DPD independently, using a simple assumption, i.e., contribution of flows from premium users to DTPD is negligible. The shape parameter of DTPD is estimated with the maximal likelihood estimation (MLE), using the given parameters α_1 and β_1. Note that the estimation process requires numerical calculation to solve the ML equation. See our previous work [13] for the detail of calculation. Finally, we estimate the shape parameter of DPD for flows larger than β_1 with the least square method. We note that the approximated distributions above are *not* continuous at the truncation point in theory. Thus, we cannot use the approximated distributions to derive statistics such as mean or variance. Our objective is to illustrate that the actual flow size distribution of a video sharing service can be divided into two distinct types of distribution models, DTPD and DPD.

The estimated shape parameters are $\theta_1 = 0.008$ for DTPD and $\theta_2 = 2.76$ for DPD. The bottom-right graph of Fig. 3 shows the results. Notice both DTPD and DPD approximate the distribution well. In addition, notice that the shape parameter of DTPD takes extremal values, i.e., $\theta \sim 0.008 < 1$ indicates that the first and second moments could be divergent if there is no constraint. In fact, more than 10% of flows is larger than 10 MB. We conjecture that these skewed parameters reflect the effect of the constraints. That is, many free membership users who hope to upload large files need to *compress* or *divide* the files so that they fit into the service capacity. Accordingly, many files that were originally larger than the limitation are made smaller than the limitation; thus, they show the *truncation* property. We note that flow size distribution and file size distribution are not exactly the same because the former reflects the popularity of file accesses.

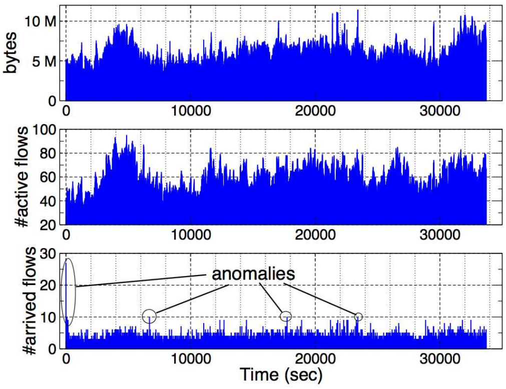

Fig. 4. Time-series of traffic volume (top), number of active flows (middle), and number of arrival flows (bottom), for traffic flows originating from YouTube servers.

However, the characteristics of flow size distribution should be correlated with the flow size distribution because a flow basically originates from a file.

In summary, we found that the flow size distributions of large-scale video sharing services exhibit *significant heavy-tailedness* and the *truncation* property, which are quite different from the property of conventional web traffic flows.

4.3 Temporal characteristics

Finally, we focus on the temporal characteristics of video flows. Understanding the temporal characteristics such as time variability and flow arrival process is crucial in building realistic traffic model. Since our data set consists of 9.5-hours of traffic data, we cannot study the cyclic patterns of traffic, i.e., diurnal or weekly variation as shown in [15]. However, as we shall see shortly, the multiple time-scale analysis enables us to explore the temporal structure of video sharing traffic. Figure 4 shows the time-series of (1) total traffic volume, (2) number of active flows, and (3) number of arrival flows, for traffic flows originating from YouTube servers. We see that the traffic volume and number of active flows are positively correlated (correlation coefficient is more than 0.8). In contrast, the number of arrival flows is independent of these two (correlation coefficient is less than 0.05).

Figure 5 shows auto-correlation functions for the time-series of total traffic volume, number of active flows, and number of arrival flows. While traffic volume and number of active flows exhibit a long-range dependence (LRD) property, the number of arrival flows does *not* have the time correlation structure, i.e., it exhibits the *memorylessness* property. These observations can be explained by the traditional traffic source model, such as that in Ref. [17]; i.e., aggregation of heavy-tailed source traffic (i.e., flows with heavy-tailed size distribution) exhibits the LRD characteristics.

Finally, we aim to validate the assumption that flow arrival process can be modeled with the Poisson model. Figure 6 shows the probability mass function of the number of arrival flows per a time unit (1 sec) in normal and log scales. We also plot the approximated distribution with the Poisson distribution model. While the approximation

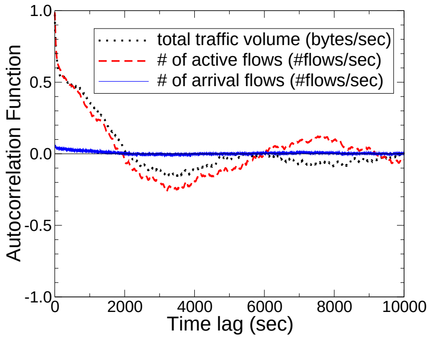

Fig. 5. Auto-correlation functions for traffic volume, number of active flows, and number of arrival flows.

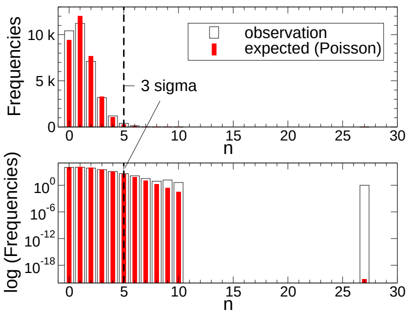

Fig. 6. Probability mass function of the number of arrival flows per sec: normal-scale (top) and log-scale (bottom).

works well over the several orders of magnitudes, we observe a small number of outliers, e.g., $n \geq 10$. For instance, we observe an event $n = 27$, which means 27 distinct flows are observed in a second. According to the Poisson approximation, the expected probability that the event occurs should be less than 10^{-18}, which is unlikely to happen in the 9.5-hours measurement period. Thus, it is likely that these extremal congestion periods are exceptional. On the basis of careful examination of the flows that constitute these outliers, we conjecture that these outliers are associated with the *flash-crowd effect* because the observed client IP addresses are not identical during the time periods.

We also applied Pearson's chi-square test to make our observation conclusive. Note that the outliers were removed before applying the statistical test. We tested a null hypothesis that the observed distribution was identical to the Poisson distribution. We concluded that we cannot reject the null hypothesis at the 0.05 level of significance.

From these observations, we may conclude that (1) the aggregated traffic of video sharing services can be modeled with the LRD traffic model and, (2) after removing outliers, the flow arrival process of YouTube can be well modeled with the Poisson arrival process. We validated that these observations hold for the other services as well and omit the results due to the space limitation.

5 Related work

This section reviews prior studies also on the large-scale video sharing services and compares them to ours. Recently, many researchers have focused on characterizing the workload of large-scale video sharing services [1–4, 7, 8, 11, 15, 20]. Huang et al. [8] analyzed the access log of MSN Video [14] and found 95% of accesses could have been eliminated by using peer-assisted content delivery system. Cha et al. [2] analyzed the properties of video files on YouTube and derived similar implications. Cheng et al. [3, 4] crawled the YouTube site and found that statistics such as length, access patterns, growth trend, and active life span were quite different compared to traditional video streaming applications. Kang et al. [11] measured and analyzed Yahoo! Video sites [18] to characterize workload of the video sharing service. Based upon obtained characteristics, they gave guideline for SLA and workload scheduling schemes on the resource management efficiency of an online video data center. Abhari and Soraya [1] investigated YouTube popularity distribution and access patterns through analysis of a vast amount of data collected by crawling YouTube API. On the basis of the observations, they presented essential elements of workload generator that can be used for benchmarking caching mechanisms.

While the above works were attempted from video sharing service provider perspective, the following works were oriented for network service providers, like ours. Zink et al. [20] analyzed YouTube traffic at a campus network and analyzed the *local* popularity characteristics of video files. In Ref. [7], Gill et al. investigated the statistics of user sessions (i.e., flows) on YouTube. They showed YouTube users transfer more data and have longer *think* times than traditional Web workloads. Their observation on file transfer volume of YouTube is coherent with our findings.

Note that both the refs [20] and [7] rely on deep packet inspection (DPI) for their analysis. In general, employing DPI exploits payload information and has been prone to privacy problem. In contrast, our method, which bases on naming/addressing conventions of large-scale server farms, does not require any payload information nor IP addresses of end-users. It just leverages *server-side* IP address, which is publicly available information. Also, thanks to its simplicity, the detection method is light-weight and scalable, while conventional DPI requires expensive processing, e.g., wire rate byte stream matching with stateful inspection of all incoming flows on high-speed links; thus employing DPI at high-speed links is a difficult task.

Plissonneau et al. [15] characterized the impact of YouTube traffic on a French regional ADSL point of presence. They revealed that YouTube video transfers are faster and larger than other large Web transfers. Their observations agree with our study on YouTube and the other video services as well. They also revealed that performance of video transfers and network load on the underlying ADSL network platform are correlated. In analysing the data set, they proposed a technique of detecting YouTube video traffic by looking up RDNS and commercial Geo-IP database with some heuristics. We note that our detection method is more comprehensive in extracting IP address blocks operated by service providers, and can be seen as a generalization of their approach.

We finally note that the originality of our work lies in the three contributions: (1) proposing a simple and privacy-friendly way of identifying video flows, (2) investigating multiple large-scale video sharing services simultaneously, and (3) traffic analysis from the perspective of network service providers, including the temporal analysis of traffic.

6 Conclusion and Future Work

In this work, we attempted to characterize traffic originating from large-scale video sharing services from the perspective of a network service provider. We presented a simple methodology that enables us to identify video flows by correlating web cache server logs and network measurement data. The key idea behind our approach is to leverage the addressing/naming conventions used in large-scale server farms. We analyzed the characteristics of the resulting classified video flows and revealed that flows originating from current large-scale video sharing services have intrinsic characteristics, the *significant heavy-tailedness* and the *truncation property*, which have not been observed in existing web traffic models. The origin of these characteristics is rooted in the differentiated service provided for free and premium membership. We also investigated the temporal characteristics of video flows and revealed that flow arrival process can be modeled with the Poisson arrival process and temporal variability exhibit LRD characteristics. Our future work includes in-depth analysis of the new distribution model. We will study empirically how the model fits the actual data. We also aim to understand the mechanism that governs the distributions and its implications on network resource management.

References

1. A. Abhari and M. Soraya. Workload Generation for YouTube. *Multimedia Tools and Applications journal*, June 2009.
2. M. Cha, H. Kwak, P. Rodriguez, Y. Y. Ahn, and S. Moon. I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pages 1–14, 2007.
3. X. Cheng, C. Dale, and J. Liu. Understanding the Characteristics of Internet Short Video Sharing: YouTube as a Case Study. *CoRR*, abs/0707.3670, 2007.
4. X. Cheng, C. Dale, and J. Liu. Statistics and Social Network of YouTube Videos. In *IWQoS 2008*, pages 229–238, 2008.
5. Cisco Systems, Inc. Cisco Visual Networking Index – Forecast and Methodology, 2007–2012. `http://newsroom.cisco.com/dlls/2008/ekits/Cisco_Visual_Networking_Index_061608.pdf`, June 2008.
6. Dailymotion. `http://www.dailymotion.com`.
7. P. Gill, M. Arlitt, Z. Li, and A. Mahanti. Characterizing User Sessions on YouTube. In *Fifteenth Annual Multimedia Computing and Networking Conference (MMCN)*, 2008.
8. C. Huang, J. Li, and K. W. Ross. Can Internet Video-on-Demand Be Profitable? In *ACM SIGCOMM 2007*, pages 133–144, Aug. 2007.
9. C. Huang, A. Wang, J. Li, and K. W. Ross. Measuring and Evaluating Large-scale CDNs. In *Microsoft Research Technical Report MSR-TR-2008-106*, 2008.
10. IRCache project. `http://www.ircache.net`.
11. X. Kang, H. Zhang, G. Jiang, H. Chen, X. Meng, and K. Yoshihira. Measurement, Modeling, and Analysis of Internet Video Sharing Site Workload: A Case Study. In *Proceedings of IEEE International Conference on Web Services*, pages 278–285, 2008.
12. Megavideo. `http://www.megavideo.com`.
13. T. Mori, T. Takine, J. Pan, R. Kawahara, M. Uchida, and S. Goto. Identifying Heavy-Hitter Flows from Sampled Flow Statistics. *IEICE Transactions*, 90-B(11):3061–3072, 2007.
14. MSN Video. `http://video.msn.com`.
15. L. Plissonneau, T. En-Najjary, and G. Urvoy-Keller. Revisiting Web Traffic from a DSL Provider Perspective: the Case of YouTube. In *Proceedings of the 19th ITC Specialist Seminar*, Oct 2008.
16. Smiley Videos. `http://www.nicovideo.jp`.
17. W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: statistical analysis of ethernet lan traffic at the source level. *IEEE/ACM Trans. Netw.*, 5(1):71–86, 1997.
18. Yahoo! Video. `http://video.yahoo.com/`.
19. YouTube. `http://www.youtube.com`.
20. M. Zink, K. Suh, Y. Gu, and J. Kurose. Characteristics of YouTube Network Traffic at a Campus Network – Measurements, Models, and Implications. *Comput. Netw.*, 53(4):501–514, 2009.