Social networks, YouTube, ADSL measurements, Capacity evaluation.

REVISITING WEB TRAFFIC FROM A DSL PROVIDER PERSPERCTIVE: THE CASE OF YOUTUBE

Louis PLISSONNEAU[*], Taoufik EN-NAJJARY[†] ,Guillaume URVOY-KELLER[‡]

Video oriented social networks like YouTube have altered the characteristics of Web traffic, as video transfers are carried over the legacy http port 80 using flash technology from Adobe. In this paper, we characterize the impact of YouTube traffic on an ADSL platform of a major ISP in France, connecting about 20,000 users. YouTube is a popular application as about 30% of the users have used this service over the period of observation.

We first observe that YouTube video transfers are faster and larger than other large Web transfers in general. We relate the throughput performance of YouTube Web transfers to the larger capacity of YouTube streaming servers, even though the distribution strategy of YouTube is apparently to cap the throughput of a transfer to a maximum value of approximately 1.25 Mbits/s. We further focus on the cases where the throughputs of YouTube transfers is lower than the playback rate of the video. We relate the bad performance of those transfers to the load on the ADSL platform, thus excluding other root causes like congestion between YouTube streaming servers and the ADSL platform.

Secondly, we focus on YouTube users' behaviors. We have discovered that about 40% of the video transfers were aborted by the client while in 19% of the cases, the client was performing at least one jump action while viewing the video. We show that abortions are only weakly correlated with the throughput achieved during the video transfers, which suggests that the main reason behind a video viewing abortions is the lack of interest for the content, rather than low network throughputs.

1. Introduction

Online Social Networks have become the most popular sites on the Internet, and this allows a large scale study of characteristics of social network graphs [12, 2, 3, 10].

The social networking aspect of the new generation video sharing sites like YouTube and its competitors is the key driving force toward this success, as it provides powerful means of sharing,

[*]Orange Labs, Sophia-Antipolis, France, `Louis.Plissonneau@orange-ftgroup.com`
[†]Eurecom, Sophia-Antipolis, France, `Taoufik.En-Najjary@eurecom.fr`
[‡]Eurecom, Sophia-Antipolis, France, `Guillaume.Urvoy@eurecom.fr`

organizing and finding contents. Understanding the features of YouTube and similar video sharing is crucial to their development and to network engineering.

Recently, YouTube has attracted a lot of attention [5, 11, 4, 9], as it is believed to comprise approximately 20% of all HTTP traffic, and nearly 10% of all traffic in the Internet [1]. Most of these studies rely on crawling for characterizing YouTube video files, popularity and referencing characteristics, and the associated graph connectivity. In [9], the authors have analyzed data traffic of a local campus network. They consider the resources consumed by YouTube traffic as well as the viewing habits of campus users and compare them to traditional Web media streaming workload characteristics.

Our work is along the line of [9] as we focus on actual video transfers from YouTube . Our perspective is however different as we consider residential users connected to an ADSL platform rather than campus users, as in [9]. Our focus is more on the performance perceived by our end users and on determining the root causes of those performance, than on in-depth characterization of YouTube usage.

Our dataset is a 35 hours packet level trace of all traffic on port 80 for our 20,000 ADSL users. Our data collection tool is lossless as compared to the one used in [9]. However, due to privacy constraints, we restricted ourseleves to capture traffic up to the TCP header. We thus do not have access to the meta-data available with YouTube which are exploited in [9].

Our main findings are the following. We first show that YouTube servers are in general much more provisioned than other Web servers servicing large contents. This discrepancy is apparently the reason that explains the significantly better throughputs achieved by YouTube video transfers compared to other large Web transfers in our data. We next focus on the cases where a YouTube transfer is apparently too slow as compared to the video playback rate. We relate the bad performance of those transfers to the load on the ADSL platform, thus excluding other root causes like a bottleneck between YouTube streaming servers and the ADSL platform. Another contribution is to show that transport level information allows to infer the state of a video transfer between a YouTube server and a client. We can thus measure the number of video transfers that are aborted by the client. This allows us to show that it is primarily the lack of interest for the content that motivates abortion of the transfer rather than a low throughput. We also observe that users tend to heavily use the jump facility provided by the Adobe flash player.

2. Dataset

We have collected the traffic of a French regional ADSL Point of Presence (PoP) over a period of 35 hours from 7:20 pm on Thursday 25th October 2007 to 6:00 am on Saturday 27th October 2007. This PoP connects 21,157 users using mainly a DSL box provided by the ISP to connect with contractual access capacities spanning from 512kb/s to 18Mb/s.

In this section, we first present how we detect YouTube video transfers. We next describe our capture tool and the database we use to derive the results in the paper. At last, we present the tool we use to extract client and server side capacity.

2.1. Detecting YouTube Video Transfers

In this section, we will discuss the identification of transfers from the videocaster. Watching a video from YouTube can be done in different ways ranging from browsing the content provider site to following a URL sent by a colleague or watching the video as an embedded object on another Web site, as YouTube offers an API to embed your favorite video in your personal home page or blog.

From a networking perspective, there is not much difference between the above methods to access a given content. In either of the cases, we observe a connection established with a front end server at the videocaster followed by a connection with a streaming server from the company.

We recognized YouTube video traffic based both on reverse DNS look-ups and using the MaxMind database (http://www.maxmind.com/). Indeed, YouTube videos can be provided either by YouTube or Google servers[§], the former being resolvable trough a reverse DNS look-up while the latter are in general not.

A practical issue we had to face was to determine if a large transfer of data from a server in the Google domain (as resolved by Maxmind) is indeed a video transfer for a YouTube video [¶]. In every case (following a URL, clicking on an embedded YouTube object on a non YouTube page or accessing the video through a YouTube Web site), we observed that a connection to a YouTube front end server was done before receiving data from the streaming servers. For every large transfer from a Google machine, we thus checked if the latter was following a connection from the same ADSL IP address to a YouTube server. If it is so, we conclude that this large Google transfer is a video transfer from YouTube.

Using the previous strategy and a threshold of one minute for the look-up of the YouTube connection, we found that all Google transfers of more than 500 kBytes were following YouTube transfers. We chose a threshold of one minute since, as explained in Section 5.1, if the user jumps in the video, a new TCP connection is set up with the streaming server without further interaction with a YouTube server. The fact that all video transfers from Google are initiated after a connection to the YouTube site suggests that the Google video web site is not popular any more, at least on our ADSL platform.

2.2. Capture Description

During 35 hours, users downloaded 1.67 TB of data on port 80. Our capture tool cuts the packets just after the TCP headers and the trace is instantaneously anonymised: the size of our trace in equivalent tcpdump format is 430 GBytes. We show the distribution of volumes (per period of 5 minutes) of HTTP traffic and YouTube traffic downloaded by the clients in Figure 1. The diurnal pattern observed in the trace is characteristic of human activities, as compared to p2p traffic whose volume tends to be more stable over time, e.g. [15, 14]. YouTube traffic accounts for 203 GBytes, i.e., about 12% of the overall port 80 traffic.

From Figure 1, we select 3 charateristic periods for the use of Web:

period A: A high activity period corresponding to the evening of the first day (25^{th} October 19:20pm to 26^{th} October 0:00am);

period B: A moderate activity period from the morning (26^{th} October 9:20) to the end of the afternoon of the second day (26^{th} October 17:20pm);

period C: A high activity period corresponding to the 2^{nd} evening of the trace (from 26^{th} October 17:20pm to 27^{th} October 0:15pm).

[§][9] reported that YouTube transfers could be served by the LimeLight CDN: this scenario was negligible in our dataset.
[¶]Another suspect could be Picasaweb that also allows flash transfers of videos.

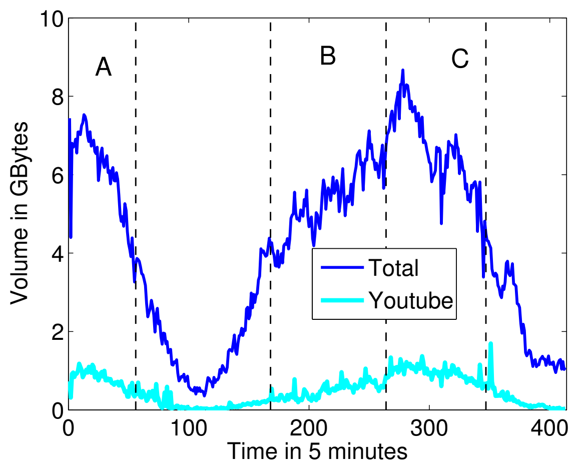


Figure 1: Volume breakdown: Total traffic vs. YouTube traffic over 35 hours

2.3. Database Description

We use a MySQL database to manipulate meta-data concerning each connection: connection identifier, volumes exchanges, throughput, RTT, packet size, reverse-lookup answer and maxmind information, etc. . . As we declare that a large transfer originating from a Google server is a YouTube video transfer if we observe a connection to a YouTube server prior to this transfer, we end up uploading in the database information about all connections on port 80 of size larger than 500 KB plus all connections from YouTube that are shorter than 500 KB. We have a total of 264,700 long connections in the database, out of which 45,563 are YouTube tranfers. Those YouTube transfers were served by 1,683 servers to 6,085 clients on the platform. The distribution of the number of YouTube transfers per client is given in Figure 2. We distinguish in Figure 2 between transfers in general and transfers that correspond to complete downloads of videos (see Section 5.1 for details). The main message from Figure 2 is that the majority of the clients view only a handful of videos while some are apparently heavy-hitters. We have also introducted in our database the client and server capacity evaluation of each connection (see section 2.4).

2.4. Client and Server Side Capacity Estimation

Since actual capacity may differ from contractual capacity due to attenuation of the line between the customer premise equipments and the DSLAM, we estimate the download capacity of users using a passive capacity estimation tool called PPrate[7]. PPrate is designed to estimate the path capacity from packet inter-arrival times extracted from a TCP connection. We use PPrate as it presents the best compromise among all available passive estimation tools to date (see [8] for detailed comparison).

In PPrate algorithm, the packet inter-arrival times are seen as a time series of packet pair dispersions, which are used to form the bandwidth distribution of the path. As this distribution is multimodal in general, the challenge is to select the mode corresponding to the path capacity. To do so, PPrate estimates first a lower bound of the capacity, and selects as the capacity mode, the strongest and narrowest mode among those larger than the estimated lower bound. The intuition behind this method is that the peak corresponding to the capacity should be one of the dominant peaks of the distribution. Note that the strategy used by PPrate is similar to the one used in Pathrate [6], a popular

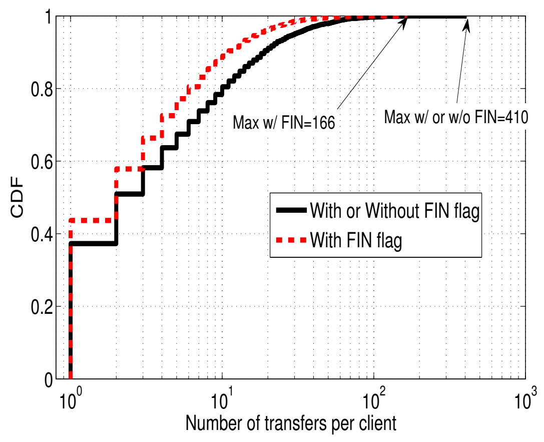


Figure 2: CDF of number of download per client

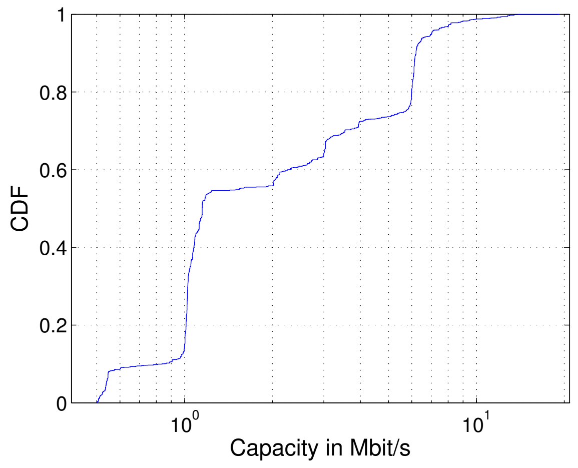


Figure 3: CDF of the Capacity of the ADSL Clients

active capacity estimation tool. More details on the exact algorithms of PPrate, and its comparison with Pathrate, can be found in [7].

Applying PPrate on the TCP data stream received from the HTTP servers, we estimate the capacity of non YouTube and YouTube servers. More precisely, we estimate the capacity of the path between a server and an ADSL client. However, since a lot of Web servers have high speed access to the Internet, and since the core of the Internet is well provisioned, the capacity of the path is in general constrained by the capacity of the server and this is what is measured.

2.4.1. Clients Capacity

In order to get consistent data, we apply PPrate only to ADSL clients having at least 3 YouTube transfers. We consider a capacity estimation for a client as reliable if the various capacity estimates are within 20% of their median value. Figure 2 reveals that about 45% of the clients perform at least 3 YouTube transfers. We eventually obtained a reliable estimation for approximately 30% of the clients.

Figure 3 depicts the capacity estimated by PPrate. We observe peaks at values close to 500 kbits/s, 1 Mbits/s and 6 Mbits/s, which is in concordance with the commercial offers made by the ISP.

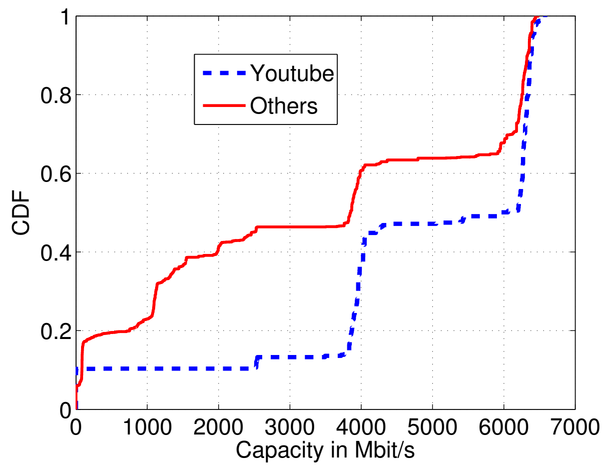


Figure 4: CDF of Servers Capacities: YouTube vs. Others

2.4.2. Servers Capacity

Figure 4 presents the servers' capacities obtained from PPrate. We observe that servers (YouTube and others) are generaly well provisioned (80% of servers have capacities larger than 1Gbit/s). Figure 4 shows that even if Web servers in general are well provisioned, YouTube servers have better access capacity with a significant amount of capacities larger than 4 Gbits/s.

Note that for approximately 5% of servers, PPrate returns abnormally low capacity estimates, i.e. values around 1 Mbit/s. We suspect that PPrate mistakenly picked a peak in the histogram corresponding to the client activity and not to the server activity. Indeed, the estimation technique of PPrate is based on modes in the histogram of inter-arrival times between consecutive TCP packets. Due to the self-clocking nature of TCP, modes exists at values close to the access capacity of the DSL client that the server is currently serving while other modes are closer to the access capacity of the server. It is out of the scope of the paper to detail the PPrate algorithm for choosing the capacity mode (see [7] for details), but in the case of high speed server serving a low speed client, we observed that if an estimation error occurs, it can lead to a severe underestimation of the capacity (overestimation is less likely to occur).

3. Large Web transfers

3.1. Global characteristics

In Figures 5 and 6, we depict the distributions of volumes and throughputs of YouTube transfers against other Web transfers for connections of more than 500 kBytes. The objective is to see how YouTube alters the characteristics of large Web transfers in the Internet.

From Figure 5, we observe that 90% of YouTube transfers are larger than non-YouTube ones. As for the throughput (Figure 6), about 70% of YouTube transfers are faster than non-YouTube ones. This is in concordance with the capacity estimation of the servers (Figure 4).

Overall, we observe that the characteristics of YouTube traffic significantly differ from other Web traffic (for transfers more than 500 kBytes). We could have expected a higher discrepancy since there is a lot of other video transfers in the remaining Web traffic (like Dailymotion or some content

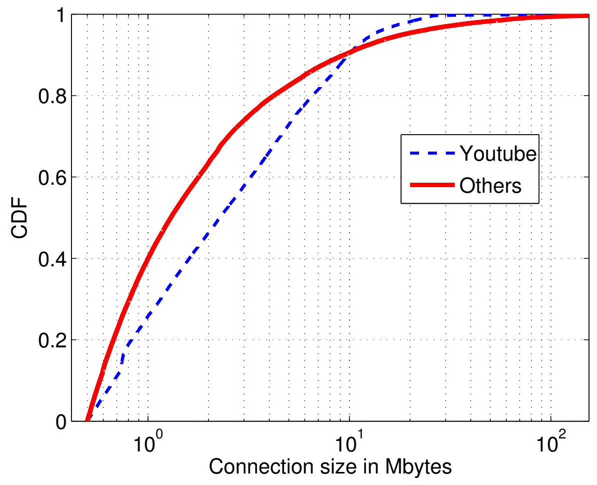


Figure 5: CDF of the Size of YouTube Transfers vs. Others

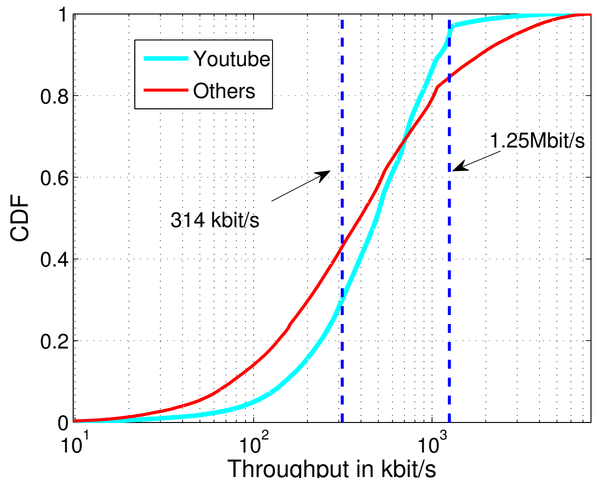


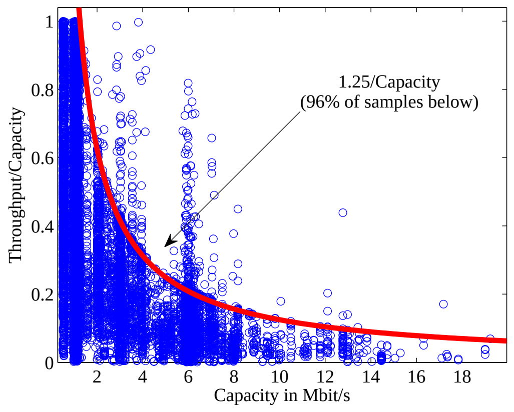
Figure 6: CDF of the Throughput of YouTube Transfers vs. Others

distribution networks like Akamai).

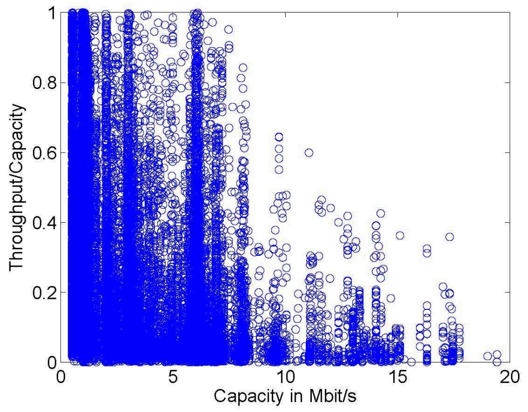
3.2. YouTube Distribution Policy

To understand the distribution strategy of web services, we depict in Figures 7(a) and 7(b) scatterplots of the ratio of the achieved throughput T of a transfer over the clients' capacity C versus the clients' capacity for YouTube and non-YouTube connections respectively. We compute these ratios for connections with more than 500 kBytes of data only and for clients for which PPrate was able to estimate the downlink capacity. A ratio close to 1 indicates that the throughput of the transfer is close to the downlink capacity of the client.

Figures 7(a) and 7(b) show very different characteristics for transfers from YouTube and from non-YouTube servers. For YouTube connections, this ratio is smaller than a specific throughput over capacity ratio in about 96% of the cases. For each transfer x, $\frac{C(x)}{T(x)} \leq \frac{K}{T(x)}$. At the limit: $\frac{C_{max}}{T(x)} = \frac{K}{T(x)} \Rightarrow C_{max} = K$. Figure 7(a) gives an approximation of the maximum throughput for a given



(a) YouTube transfers



(b) Non-YouTube transfers

Figure 7: Scatterplot of Achieved Throughput over Estimated Capacity vs. Estimated Capacity

YouTube transfer of $C_{max} \approx 1.25\ Mbits/s$. Note that half of our clients have an access capacity smaller than this threshold (see Fig. 3). Even if we restrict to clients with capacity larger than 1.25 Mbits/s, we still obtain that 92% of them are below the above asymptote. Such a distribution strategy makes sense as the playback rate of 97% of the videos is below 1 Mbits/s [9]. In addition, it can prevent ADSL or cable clients with high capacity to consume too much of the capacity of YouTube data centers.

No such throughput limitation is clearly visible for other large Web transfers, as can be seen in Figure 7(b). We simply observe that the higher the capacity of the client, the less likely it is for the transfer to saturate the downlink. One could argue that the latter result is understandable as other large Web transfers come from a variety of Web servers under the control of widely different organizations. However, what Figure 7(b) demonstrates is that the phenomenon observed for YouTube is not an artefact of the ADSL platform we consider in this study.

4. Troubleshooting User's Performance

In this section, we investigate the root cause of the low throughputs of some YouTube transfers. Indeed, we observe from Figure 6 that about 30% of YouTube transfers have a throughput lower than 314 kbit/s. Display of the videos corresponding to those tranfers might lead to periods where the video is frozen, as the vast majority of videos have bitrates between 300 and 400 kbits/s (see [9], Figure 10). Figure 7(a) further suggests that bad performance can occur even to all clients, almost irrespectively of their access capacity.

Several factors can explain why a given TCP transfer achieves a given TCP throughput: the application on top of TCP or factors that affect the loss rate or the RTT of the transfer, as highlighed by the TCP throughput formula [13]. Similarly to the approach followed in [16], we checked on a few example transfers that the application (flash server) was not responsible for the observed low throughput. Indeed, the fraction of push flag (set by the application to indicate that there is no more data to transfer) is negligible for YouTube transfers and packets inter-spacing is never commensurate to the RTT of the connection. It is demonstrated in [16] that those two effects are the two possible footprints left by the application at the packet level.

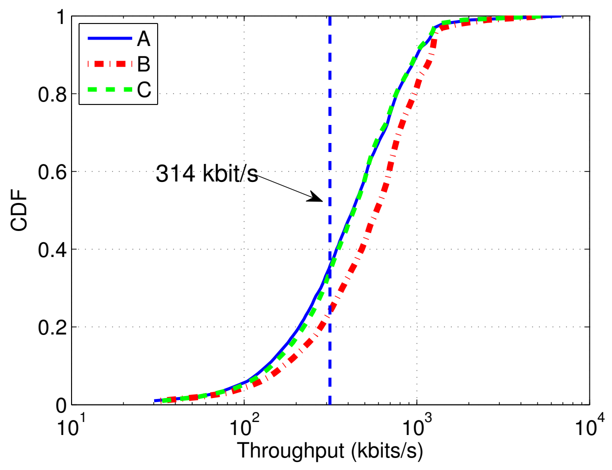


Figure 8: CDF of the achieved throughput for the three periods

We thus conclude that the application is not responsible for the slow transfers. We next focus on the loss rate or the RTT of the transfers. Our intuition is that this is the load on the ADSL platform that explains the low transfer rates we observe. To prove our intuition, we make use of the three periods A,B and C defined in Section 2.2: periods A and C correspond to evenings, and are more loaded than period B that corresponds to the middle of the day. In Figure 8, we observe that the higher throughputs occur in period B, i.e., during the period where the ADSL users generate the lowest amount of Web traffic, which suggests a correlation between the local load and the troughput of the YouTube transfers.

We next focus on losses. The metric we consider, as an estimate of the loss rate, is the fraction of retransmitted packets. In Fig. 9, the retransmission rate is indeed much lower for period B (Note that as we use a logarithmic scale, the mass at zero is not directly visible but corresponds the onset of the curves).

This discrepancy between the periods, in terms of losses, might explain the lower throughputs we observe, but we would like to understand if they are due to a local or distant congestion. We thus consider the RTTs, as increasing loss rates are often correlated with increasing RTT. Our measurement probe, which is located close to the ADSL users enables us to compute the local RTT, between the probe and the ADSL host and the distant RTT between the probe and the YouTube server. Figure 10 (resp. 11) depicts the distant (resp. local) RTT for all YouTube transfers. We observe from Figures 11 that the most likely cause to explain performance degradation of YouTube transfers is an increase of the local RTT, i.e. an increase of the local load. In contrast, distant RTTs - see Figure and 10- seem unaffected by the exact time period one considers.

As a conclusion, we observe that the performance of YouTube transfers is apparently correlated with the local load of the ADSL platform.

5. User behavior

In this section, we present our findings regarding the way users watch videos and its impact on the network traffic. A lot of works, e.g. [4], advocate the use of caching for YouTube and other social

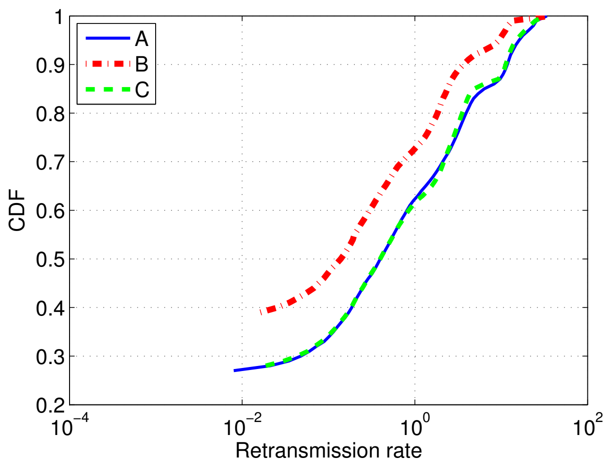


Figure 9: CDF of the retransmission rate for the three periods

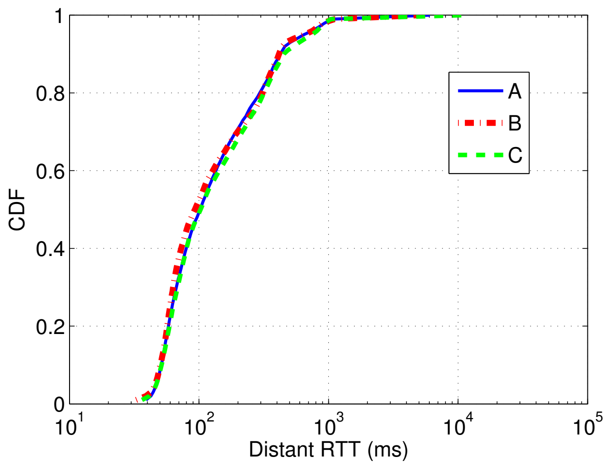


Figure 10: CDF of the distant RTT for the three periods

network traffic. However, while the popularity of videos is displayed by YouTube on its page[||], little is known about the way users watch those videos, and especially if they watch the full video or not. In the extreme case, we could have a popular video that is very long with users watching only the beginning of the video. In this case, caching the full video does not necessarily make sense.

For the above case, we have first performed experimental transfers to assess the state of a video connection and we have then applied our findings to our ADSL trace.

5.1. Assessing the state of a video transfer

Here, we focus on the termination of TCP connections induced by video transfers. Indeed we expect that the end of a TCP connection from the streaming server reflects the status of the file transfer from a user's perspective. We consider the following user behaviors:

[||]See `http://www.masternewmedia.org/news/2008/02/29/internet_video_metrics_when_a.htm` for some details about how videocasters count an actual view of the video.

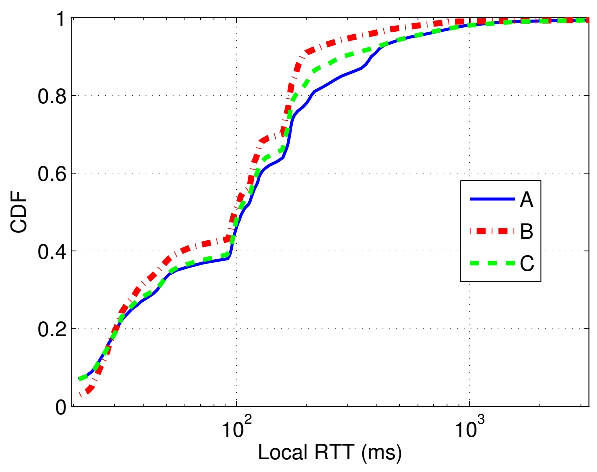


Figure 11: CDF of the Local RTT for the three periods

- Watching a whole video;
- Jumping in a video;
- Switching from one video to another;
- Closing the browser while watching a video.

For the above user actions, we have tested the behavior of the servers using Firefox (Version 2.0.0.12) and Internet Explorer (Version 7.0.5730.13) on Windows XP SP2. The first step of our analysis was to extract the TCP connection corresponding to the actual video transfer from the others. In the case of YouTube, we look for the message `HTTP:GET /get_video?video_id=` $\cdots$ In some cases, a message
`HTTP:GET /videodownload?secureurl=` $\cdots$ occurs for a video transfer.

At the server side, we have observed only two different types of activity for all user behaviors we have considered. If the server completes the transfer of all data associated with the video, its last data packet will carry the `FIN, PSH` and `ACK` flags. If not, the server will continue sending data as long as it has not received a RST from the client. It then abruptly stops sending data. This also holds true for a jump action that results in the opening of new TCP connection with the same streaming server.

From the client perspective, we observed that whenever the user closes the browser/tab or switches to another video or jumps in the same video, Internet Explorer sends a RST followed by additional RSTs for every new arriving packet from the server. Firefox behaves slightly differently in that it first sends a FIN packet, that is acknowledged by the server but somehow gets ignored as the latter continues sending data that trigger RSTs at the sender side. Eventually whenever RST are received at the server side, the transfer gets aborted.

From the above analysis, we decided to categorize YouTube video transfers into two sets. The first set corresponding to a completed connection with a `FIN` sent by the server and the second set corresponding to partial viewings of videos, either because of the abortion of the viewing or because of a jump action.

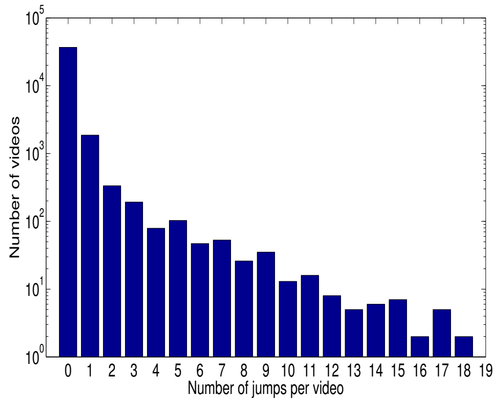


Figure 12: Histogram of jump actions

5.2. When is a view a view?

Using the results obtained from the previous section, we estimated the number of video transfers for which there was a single data transfer between a YouTube server and a client. Over a total of about 45,000 interactions between a streaming server and a client, 36,700 correspond to a single connection while the rest correspond to multiple connections between the client and the same streaming server, i.e. jumps while viewing the video. Note that the actual number of jump actions might be higher than the one we observe as we can detect a jump only if it has an impact on the transfer on the wire. However, in a number of cases, and especially if the video is short, it is likely that the video will be fully received before the jump is performed. In this case, this action is handled by the flash player and has no effect on the network traffic that we could measure.

Let us first concentrate on the 36,700 cases where there is a single transfer. In half of the cases only, the connection terminates correctly while in the other cases, the user stops the transfer either because of a lack of interest in the content or because of bad network conditions. We address this question in Section 5.3.

As for the jump action, we present in Figure 12 the distribution of the number of jumps per video. In 93% of the cases, there is single jump action (note the logarithmic scale of the figure). Not suprisingly, the larger the number of jumps, the smaller the number of samples.

# of jump actions	Mean Volume in Mbytes
0	4.74
1	4.77
2	6.06
3	6.28
4	6.37
5	6.97
6	8.16
7	7.77
8	8.94
9 or more	15.20

Table 1: Mean Volume of Transfers vs. Number of jump actions

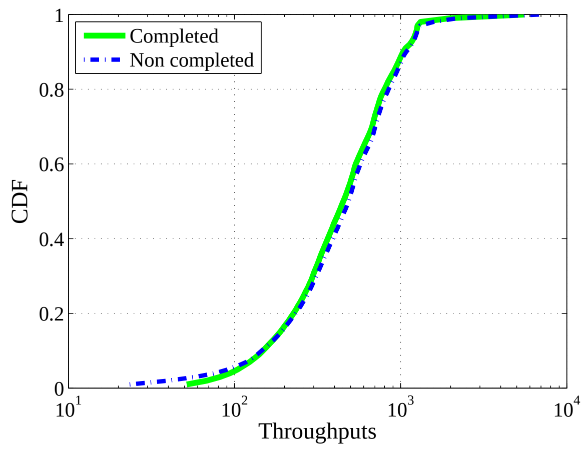


Figure 13: Througputs for completed and non completed video transfers

We have computed - see Table 1 - the total amount of data exchanged between a given client and a given YouTube server, depending on the number of jump actions taken. Table 1 shows that the video transfers with jump actions are much longer than others. Indeed, to be able to jump in a video before completing its download, it must be long enough.

5.3. Video transfer status vs. throughput?

In this section, we focus on the correlation between the status of a video transfer (completed or not), and its throughput. Indeed, there might be two possible causes for the abortion of a transfer: a lack of interest for the content or a too low transfer rate. To investigate this issue, we plot in Figure 13 the throughput of completed video transfers and non completed video transfers. We observe that even if a few percents of the non completed transfers have lower throughputs than completed ones, throughputs in both categories are similar. This suggests that a low throughput is not the primary reason for aborting a video transfer. This is further confirmed by the fact that about 32% of the completed video transfers were performed at rates lower than 314 kbits/s. This confirms that users are quite patient if they really want to view a content.

6. Conclusion

In this paper, we have investigated the characteristics of YouTube traffic. Our main findings can be summarized as follows. YouTube traffic accounts for as much as 20% of the Web traffic. The characteristics of YouTube traffic (volumes, throughputs) significantly differ from other Web traffic when considering large connections (more than 500 kBytes). YouTube servers apply a rate limitation policy for content distribution, with a maximum transfer rate of about 1.25 Mbits/s. This holds even though their servers seem to have better access capacity than an average Web Server in the Internet. As for the impact of user behavior, we have found that about half of the video transfers were aborted, probably because of lack of interest in the content or because of poor network conditions. We have also detected that users perform some jump actions in 19% of the cases.

As an extension of this work, we plan to figure out the observed characteristics of YouTube on a long term analysis, and to compare YouTube traffic to other video streaming ones. We have already established that in the analyzed data, Dailymotion and its content provider LimeLight represent a significant traffic volume in our dataset.

[1] "Ellacoya data shows web traffic overtakes Peer-to-Peers as largest percentage of bandwidth on the network", http://www.ellacoya.com/news/pdf/2007/ NXTcommEllacoyaMediaAlert.pdf.

[2] L. A. Adamic, O. Buyukkokten, , and E. Adar, "A social network caught in the Web.", In *First Monday*, 2003.

[3] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution", In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2006.

[4] 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 *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, New York, NY, USA, 2007, ACM.

[5] X. Cheng, C. Dale, , and J. Liu, "Understanding the Characteristics of Internet Short Video Sharing: YouTube as a Case Study.", arXiv:0707.3670v1, Cornell University, July 2007.

[6] C. Dovrolis, P. Ramanathan, and D. Moore, "What Do Packet Dispersion Techniques Measure?", In *Proc. of IEEE INFOCOM'01*, pp. 905–914, Los Alamitos, CA, April 2001.

[7] T. En-Najjary and G. Urvoy-Keller, "PPrate: A Passive Capacity Estimation", In *IEEE e2emon*, Vanvouver, CANADA, 2006.

[8] T. En-Najjary and G. Urvoy-Keller, "Passive Capacity Estimation: Comparison of Existing Tools", In *International Symposium on Performance Evaluation of Computer and Telecommunication Systems*, Edinburgh, UK, 2008.

[9] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge", In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 15–28, New York, NY, USA, 2007, ACM.

[10] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks.", In *Proceedings of the National Academy of Sciences (PNAS)*, pp. 7821–7826, 2002.

[11] M. J. Halvey and M. T. Keane, "Exploring social dynamics in online media sharing", In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, New York, NY, USA, 2007, ACM.

[12] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks", In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 29–42, New York, NY, USA, 2007.

[13] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP Throughput: A Simple Model and its Empirical Validation”, In *Proc. of ACM SIGCOMM’98*, pp. 303–314, Vancouver, Canada, August 1998.

[14] L. Plissonneau, J.-L. Costeux, and P. Brown, “Analysis of Peer-to-Peer Traffic on ADSL”, In *6th International Workshop on Passive and Active Network Measurement*, pp. 69–82, January 2005.

[15] M. Siekkinen, D. Collange, G. Urvoy-Keller, and E. W. Biersack, “Performance Limitations of ADSL Users: A Case Study”, In *Proc. Passive and Active Measurement: PAM 2007*, April 2007.

[16] M. Siekkinen, G. Urvoy-Keller, and E. W. Biersack, “On the Interaction Between Internet Applications and TCP”, In *Proc. 20th International Teletraffic Congress: ITC*, June 2007.