Insights of YouTube View Check System

Miriam Marciel
IMDEA Networks
Universidad Carlos III de Madrid
Leganés, Spain
Email: miriam.marciel@imdea.org

***Abstract*—YouTube is the most popular website for videos nowadays. Its business model is based on advertisement: ads are introduced in the videos and uploaders are paid based on the number of views of their videos. For this reason, it is imporant for YouTube to be able to check that views are real, and discard those that are faked.**

The main purpose of this paper is to understand the mechanisms implemented by YouTube to detect and discard faked views. To this aim, we have developed a robot that performs faked views; by building on a thorough study of the patterns and the most common traffic sources of videos, and emulating this patterns, our robot shows a very similar behavior to real users. From the results obtained from our experiments, we observe that: (i) we are able to overcome the 301 limit, which corresponds to the first thorough analysis YouTube performs to detect faked views, (ii) we are capable of pushing the total number of views to an unlimited large number, but at a limited growth rate, (iii) YouTube appears to be insensitive to many behaviors that show that views are faked, except for the IP address used by the client, and (iv) regardless of how sophisticated the robot is, YouTube only counts a fraction of the views realized (around one half).

In order to compare the behavior observed by our robot to other behaviours, we have considered (i) an experiment with real people watching the video, and (ii) services offered in the Internet that sell views for YouTube videos. We have observed from the first experiment that even with real views, YouTube only seems to be counting about half of them as real views. Furthermore, the second experiment confirms that even some basic tools are able to increase the number of views (e.g., in some cases we found that the average view duration was 0 but still for those cases YouTube counted a large number of views).

The experiments conducted seem to show that YouTube implements a very basic scheme to detect fake views that is easily tricked by simple tools, and that YouTube discards a subsantial number of the views performed even for real views. Such results seem to point to the need for revisting the algorithms implemented by YouTube to detect fake views.

***Index Terms*—YouTube, fake views, data analysis, internet measurement**

I. Introduction

YouTube is the most popular User Generated Content and Video-On-Demand site. According to the statistics provided by YouTube[1], it has more than 1 billion unique users and over 6 billion hours are watched every month. Aditionally, it is ranked in Alexa[2] as the third Top Site.

In December 2012, YouTube removed more than 2 billions of views in channels belonging to music companies such as Universal, Sony/BMG and RCA[3]. The reason whas that YouTube detected that they were making fakes views and decided to cancel views. After this incident, YouTube announced that they enforced their viewcount policy.

YouTube business model is based on advertisement in their videos. YouTube has a monetization program in which uploaders receive money depending on the number of views in their channel[4]. Therefore, it is very important for YouTube to check that the views are real.

The YouTube check system works as follows. The first time that it the views is when the video reaches 301 views. Before that threshold YouTube does not check if the views are real or not and the viewcounter is increasing in real time. When the video reaches 301 views, YouTube stops the viewcounter and checks the views, classifying them as real or fake. The counter may be frozen for several hours. After the check is done, if the views are classified are real, the viewcounter is updated. If the views are classified as fakes, the video is removed from the uploader channel and it is not possible to search the video, although it can be accessed with the direct link. The viewcounter then shows 301 views, but it is no longer updated.

With this paper, we aim at answering the following questions:

- Can we cheat YouTube?
- Can we overcome the 301 views limit?
- Can we infer what YouTube checks?

To address the issues pointed out above, we have designed and developed a robot whose purpose is to make fake views with a behaviour as natural as possible. In order to replicate the behavior of real videos with real views, we have crawled some YouTube videos and analyzed the typical statistics of the videos over time (number of views, likes, dislikes) as well as the traffic sources of the videos.

Building on the statistics gathered, we have designed and implemented a robot with different modules that implement features that we believe YouTube could check. We have then run several robots with different active modules to see which of the features are checed by YouTube. From our exepriments, we have found out that the IP address is an important factor but not necessarily critical. Additionally, YouTube does not

[1]www.youtube.com/yt/press/statistics.html
[2]www.alexa.com/topsites
[3]www.dailymail.co.uk/sciencetech/article-2254181/YouTube-wipes-billions-video-views-finding-faked-music-industry.html
[4]http://www.youtube.com/partners

check some HTTP headers that some proxies send.

In order to gain additional insights into the YouTube check system, we explored services in some crowdsourcing websites that offer fake views in YouTube in exchange of a monetary payment. We hired some of these services and found out that YouTube counts thousands of views with a view duration of 0 seconds, which means that views are counted even when YouTube knows that the users have not watched the video.

The remainder of this paper is organized as follows. In Section II presents the related work. Then, in Section III we show the analysis of the data that we collected. Section IV describes the robot and Section V presents the tests and results that we obtained with this robot. In Section VI, we describe how people do business with views. The paper then closes with Section VII, which describes the future work and provides some concluding remarks.

II. Related Work

A lot of studies have been conducted on YouTube, focusing on aspects such video popularity and user behaviour among others. For instance, Cheng *et al.* [1] studied the statistics of YouTube videos and the social network of YouTube, concluding that YouTube is a small-world.

Many studies on YouTube have specifically focused on the popularity of the videos. In [2], the authors characterize the distribution of the popularity and its evolution over time. Other sudies, such as [3], [4], [5], [6], have attempted to predict video popularity in YouTube. Figueiredo *et al.* [7] measured how the popularity of videos evolves over time and analyzed the most common traffic sources.

YouTube's recommendation system has also received some attention. Zhou *et al.* [8] studied how recommendations work in YouTube and their impact in the number of views. The authors observed a strong correlation between the viewcounter of a video and the average viewcounter of its top related video. They also observed that the position in the related video list is important.

In [9], the authors created a dataset with clone videos and studied what impacts popularity. For old videos, they observed "richer-get richer" behaviour, while for young views, they observed that the social network of the uploader, the quality of the video and the number of keywords are the factors that impact popularity.

Brodersen *et al.* [10] studied the geographic distribution of views. They found that 50% of the videos have 70% of the videos in one region. They also studied the relation between social factors (e.g., views coming from Online Social Networks) and local popularity.

Gill *et al.* [11] made some measurements in a campus network and compared the tradicional Web with the new Web 2.0; the study showed that there is a longer think user time and that more bytes are transfered. Another study from an edge network [12] compared local popularity and global popularity, concluding that there is no strong correlation. Besides, the authors also run some simulations with alternative distribution infrastucture, reducing the network traffic.

Total number of videos	60826
Videos with YouTube Insight available	6006 (9.8%)
Videos with $>$ 300 views	6530 (10.7%)

TABLE I
Dataset Information

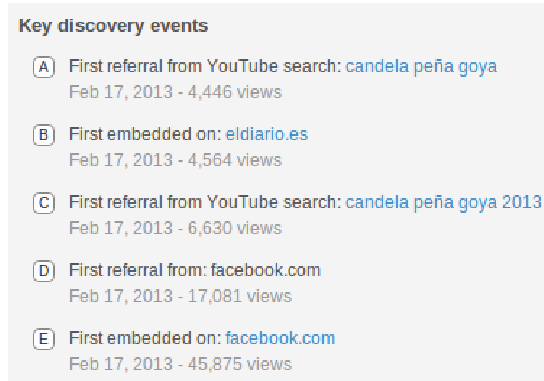


Fig. 1. Traffic sources presented by YouTube Insight

The above shows that the YouTube system has received a substantial attention from the research community, and that video popularity is very important from a number of viewpoints. To the best of our knowledge, this work is the first one to analyze the impact of fake views on video popularity and the system YouTube employs to detect them.

III. Data Analysis of YouTube

In this section we describe the analysis that we have performed of YouTube's data and report on the main results obtained from this analysis.

A. Dataset description

In order to simulate the normal behaviour of videos and thus make it more difficult for YouTube to identify the views of our videos as fakes, we analyzed the actual behavior of YouTube views. To this end, we have retrieved and studied real data from YouTube.

Our dataset has two sources of information. The first one is YouTube API. This API allows us to collect data from YouTube videos such as number of views, comments, likes, dislikes and uploader information at one instant of time. The second source of information is YouTube Insight. YouTube Insight gives us *Key discovery events*[5], which are the most common traffic sources for a video.

To build our dataset, we have collected data from 60826 videos that were uploaded on the 1st of February 2013. The information of the dataset is presented in table I. We recorded the view counter all the videos hourly during their first week. At the end of the first week, only 6006 videos had YouTube Insight information available and 6530 videos had more than 300 views.

[5] At the time of writing this paper, this information was no longer available in YouTube Insight.

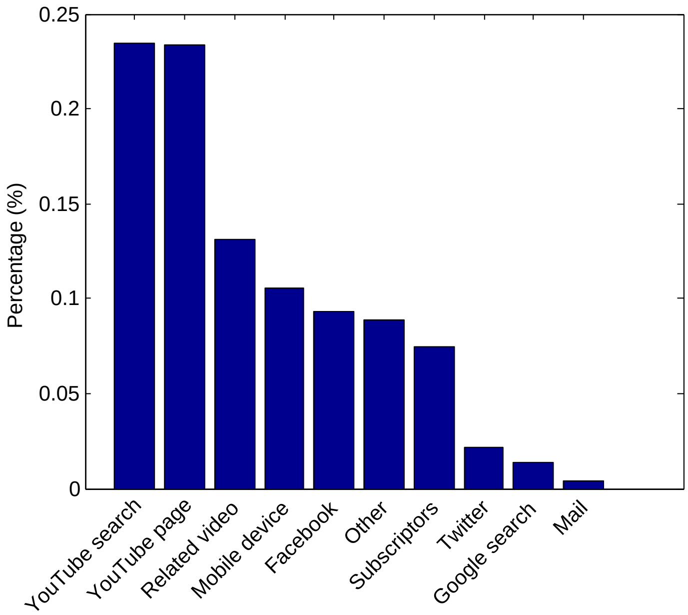


Fig. 2. Most common traffic sources in the first week

B. Analysis of events

YouTube Insight only gives the 10 most common sources of the videos (at most), and thus there may be views that cannot be captured with the information provided. Moreover, this information is not available in all the videos as the uploader has to give permission to make it public. Figure 1 shows how these events are presented; as we can see, a description of the event as well as the number of views coming from that source is provided.

In total, we classified the following traffic sources:

- YouTube search: views coming from a search in YouTube.
- Subscriptors: users watching a video from the subscriptor module in YouTube.
- Related video: requests having as referrer another video.
- YouTube page: views coming from pages inside YouTube such as the channel page, a playlist, watch later page, feature videos, etc.
- Mobile device: views coming from a mobile device.
- Facebook: views having as referrer the Online Social Network Facebook.
- Twitter: views coming from the Online Social Network Twitter.
- Google search: views coming from a search in Google.
- Mail:views having as a referrer a mail services such as Yahoo, Hotmail, Gmail or AOL.
- Other: views that do not fit in any of the categories presented, such as views coming from videos that were embedded in some websites.

In Figure 2, we show the percentage of each of these traffic sources for the collected dataset.

The three most common traffic sources are within YouTube: searches, pages inside YouTube such as channel page or playlists and views from related views. These three sources cover almost the 60% of all the events that were identified, which shows that most of the views are coming from YouTube. The views from mobile devices are an important source of traffic, as it accounts for 10% of the views. Online Social Networks are an important source of traffic, Facebook and Twitter accounting for almost 10% of the views. These results are similar to [7] and [8]; however, unlike those works, we do not have featured videos as a main source as we crawled all the videos uploaded in one day.

C. Analysis of statistics

In the following, we report the statistics obtained from the dataset resulting from the API. To obtain these statistics, we only took into account those videos that have more than 300 hundred views in their first week, as we are interested in the behavior of the videos that overcame this limit.

In figure 3, we show a boxplot with the number of views, likes or dislikes of these videos during their first week of lifetime. Notice that the axes range has been adjusted to be show the normal behavior of most of the videos. We do not present the statistics for comments, as we did not implement a module that performs automated comments in the robot.

As it can be observed from the results, most of the videos have a median of 750 views in the first week and a median of 15 likes. Dislikes are not very frequent as at the end of the first week: the median is one.

Based on the data collected, in the following we compute the probability that a user clicks like or dislike. Note that the Likes or dislikes buttons can be pressed only by signed in users. Thus, the probability of clicking these buttons is going to depend on the percentage of views made by signed in users and the number of likes. In particular, the probability of clicking like or dislike is given the average number of likes divided by the average number of views made by signed in users (the latter is given by the average number of views multiplied by the percentage of signed users we are going to have. Therefore,

$$Prob_{like} = \frac{\overline{likes}}{\overline{views} * Percentage_{users}} \tag{1}$$

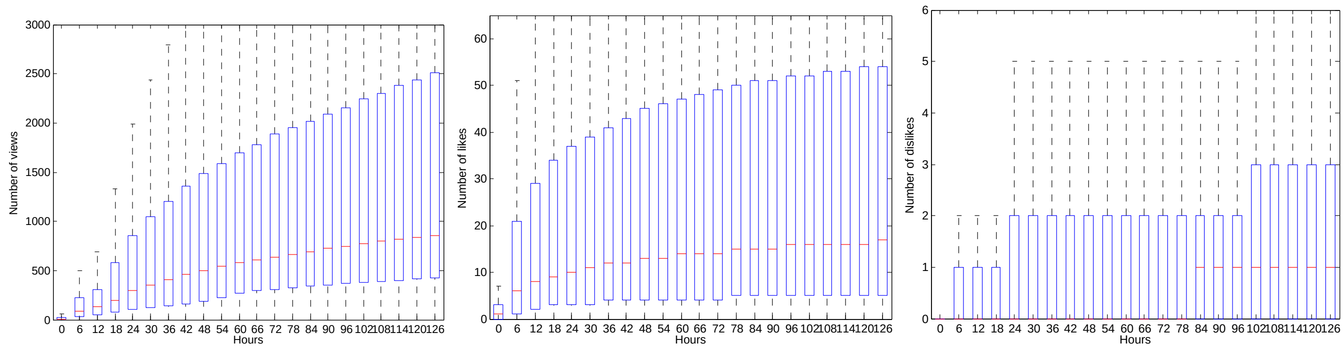
$$Prob_{dislike} = \frac{\overline{dislikes}}{\overline{views} * Percentage_{users}} \tag{2}$$

where $Percentage_{users}$ is the fraction of signed users.

To compute the above probabilities, we retrieve the number of views, likes, dislikes from our dataset, and assume that the percentage of signed users is of 50%.

IV. Performing fake views: Robot implementation

In order to understand YouTube's check system, we have implemented a robot that performs fake views while showing a behavior as close as possible to real users. One of the key features of the implementation of our robot is that it is modular; thus, by activating/deactivating some of the robot's modules, we can evaluate the sensitivity of the view counting to different behaviors.



(a) Number of views (b) Number of likes (c) Number of dislikes

Fig. 3. Evolution of the number of views/likes/dislikes during the first week

A. General Description

We have developed a robot whose purpose is to make fake views in a video. To this end, the robot makes views from different locations (IP addresses), with different user-agents, simulates views from the most common traffic sources that we have analyzed, watches the video for a certain duration and behaves in YouTube as a normal user would do, going back, closing the browser, watching a related video, etc.

The robot has a list of users that connect and watch the video. For us a user has assigned a proxy, a user-agent and it may have assigned a YouTube account if that user is signed in. We use 71 PlanetLab nodes as proxies and configure 26 user-agents with different web browsers and operating systems.

The robot was programmed using Selenium Webdriver, a library for testing web applications. This library allows us to configure parameters such as user-agent, proxy and cookies and search for elements in the HTML DOM, click them and fill up forms.

From the traffic sources that we listed in section III-B, we developed YouTube search, Facebook, Twitter, Google search and Mail, and we add via direct link. We did not implement YouTube page, related video and subscriptors as these traffic sources are generated automatically by YouTube. Mobile views were not implemented because we were not able to watch the video, as computers do not have the mobile plugin. We plan to do this as a future work using libraries that simulate mobile browsers.

In order to follow the behavior of real users, we set a weight to each of the events that is proportional to the percentages obtained in the previous section.

B. Flowchart

The robot has the flowchart shown in figure 4. A new view from one user is going to arrive with an exponential inter arrival time, creating a Poisson process for arrivals. This is the most common behaviour of videos dynamics [13].

When a view is about to start, we configure a user, either by creating a new one or retrieving an old user. For each new user, we randomly configure its proxy, its user-agent and if the user is signed in or not. These parameters are not going to change when the user is retrieved again, to avoid misconfigurations that YouTube could detect. When a user finishes watching a video, we store its cookies. When an old user connects again, we retrieve its proxy, user-agent, its cookie and if it is signed in and configure them.

After the user is configured, it chooses randomly one of the traffic sources and starts watching the video. At that point, it decides if it watches the entire video or just a part of it. In case that the user does not watch the entire video, it only watches a random time following an exponential distribution with mean the duration of the video. Additionally, if the user is signed in, it can press the like or dislike button with the probability we compute in section III-C.

After watching the video, the user decides among the following options: watching a related video, going back, refreshing the page or closing the web browser. In case of refreshing or going to a related video, the user decides again between watching the entire video or part of it.

Before closing the web browser, the robot always stores the cookie for that user so it always loads the correct cookie. Then, a new user will watch the video after waiting an exponential time.

C. Modules

The robot was implemented with different modules so that the robot can be run with different modules. We have the following modules:

1) User-agent: with this module each user has a different user-agent.
2) Proxies: the users use a PlanetLab proxy to connect. If this module it is not configured, all the views are made from the same IP address.
3) Signed in users: a percentage of views come from users that are signed in. These users are going to do like or dislike.
4) Traffic sources: the robot has different traffic sources to the video. If this module is not configured, the default traffic source is YouTube Search.
5) View duration: we can configure the amount of time that the robot watches a video.
6) Actions after video watched: when the user has finished watching the video, it watches a related video or re-

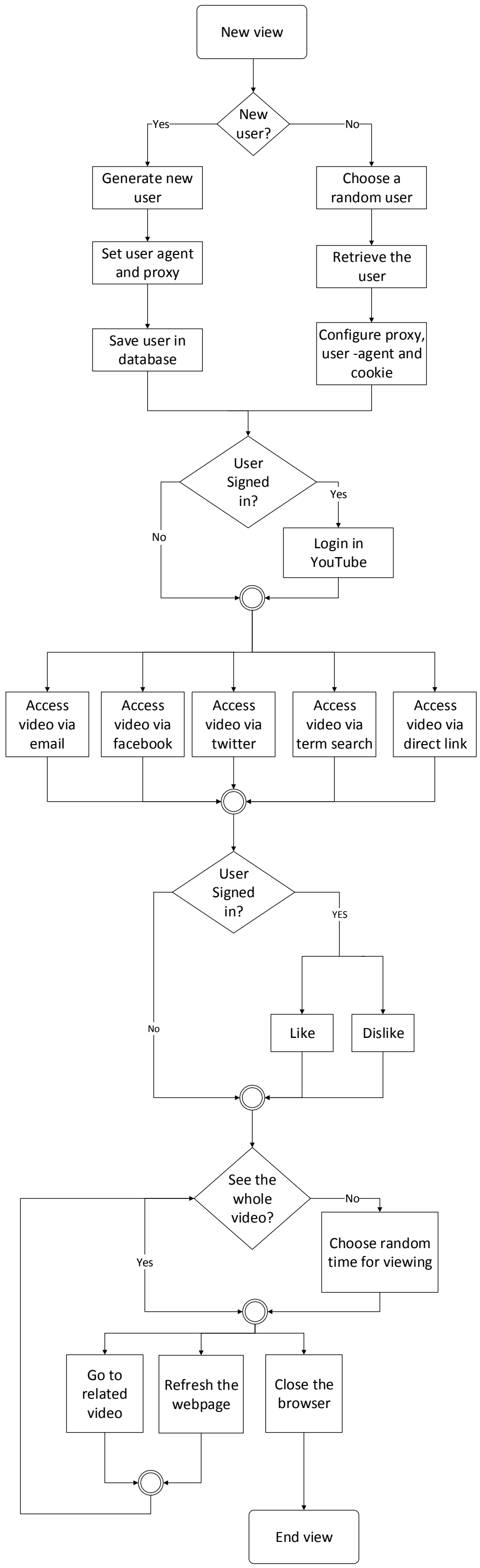


Fig. 4. Flowchart of the robot

freshes the webpage. If this module is not configure, the user closes the web browser.

7) Recovered cookies: if this module is configured, with a certain probability, we recover a user cookies and, for YouTube, it seems that the same person has watched the video again.

V. EXPERIMENTAL EVALUATION AND RESULTS

In this section, we report on the experimental evaluation conducted with the robot described in the previous section, and give the most significant results obtained.

A. Setup of all the experiments

For every test we uploaded a new video to reduce the probability of other people accessing the video. Furthermore, in order to avoid copyright issues, the videos that we uploaded were open source.

As we are the uploaders of the video, we have some video statistics that YouTube provides to the owner, namely YouTube Analytics. These statistics provide the total number of views, number of views per day, ad performance (in case of monetize is activated), views per country and per state in the US, demographics data (genre and age), traffic sources, operating system and device information, audience retention and playback location (YouTube, embedded in other websites or mobile applications). These data are updated every day.

One detail that we have to mention is that when using the PlanetLab nodes as HTTP proxies, they are sending two fields in the header: *Via* and *X-Forwarded-For*. *Via* sends the proxy that we are connected through and *X-Forwarded-For* sends our real IP address. So YouTube may know that we are connected through a proxy and check our real IP address.

B. First insights

Our first experiment was running the robot with all the modules explained in section IV-C with the aim of overcoming the 301 limit.

Figure 5 shows the accumulative number of the views of the video in the API and in Analytics per day. The views of the API were taken every day at the same time. As we can see we overcame the limit but the videocounter was stuck in 301 views for 3 days. So we can conclude that overcoming the 301 limit is easy to do.

Checking the video Analytics, we see that the locations of the views correspond to the location of the proxy, not to the location of the *X-Forwarded-For* field. Thus, YouTube does not check this field and it takes the location from the IP in the IP header.

Another insight is that most of the days there is a difference of views between YouTube Analytics and YouTube API. This could be due to the update time of the statistics YouTube Analytics, as it is only updated once a day and we do not know the interval of hours that YouTube takes as an input for the statistics.

Moreover, we realized that YouTube does not count all the views of our robot. This fact can be seen in figure 5,

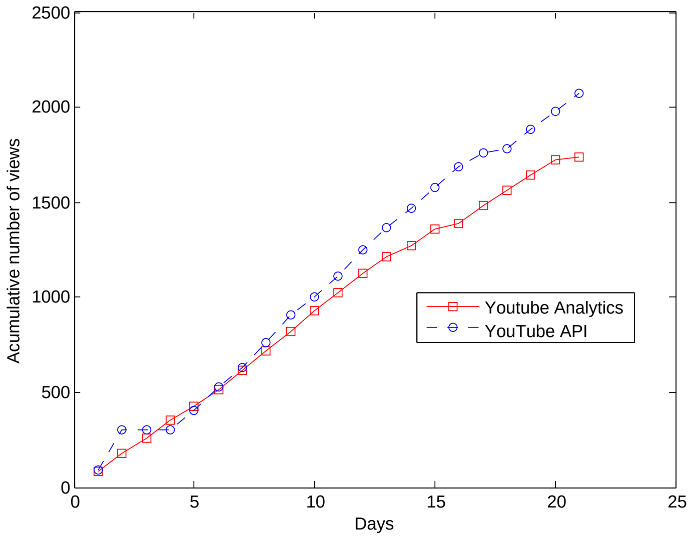


Fig. 5. Number of views of YouTube Analytics and YouTube API

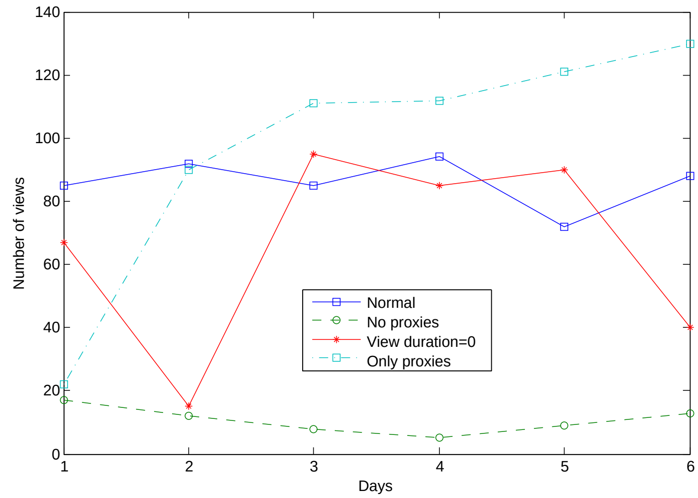


Fig. 6. Comparison of robots

before the views of YouTube Analytics reach 300 views. As commented in the introduction, before having 301 views the viewcounter is increased in real time. In the figure we can see that, in the second day, the video already has 301 views in the viewcounter, but YouTube Analytics have only counted around 200 views. Our robot is making 432 views per day in average and only 86.2 views are counted by YouTube for this robot. We will talk this subject with more detail in sections V-D and V-E.

For the rest of the paper we only consider the views of YouTube Analytics, as these ones have been checked and marked as valid and because the views are not synchronized.

C. Removing/Adding modules

In order to discover what YouTube checks, we run some robots with different modules commented in section IV-C, trying to see the differences of the views counted.

The robots that were configured are the following:

- Normal: a bot with all the modules configured.
- No proxies: this robot does not have any proxy configure, thus it makes all the views from the same IP address. The rest of modules are configured.
- Only proxies: this bot has only configured the modules of proxies, as we think is the most important module. The traffic source of the video is YouTube Search as it is the most common.
- View duration = 0: this bot does not watch the video, just loads the page of the video so the view duration is 0. The rest of the modules are configured.

All these robots have same interarrival time between views so, in average, they all make the same number of views.

We decided to configure these robots because we think that the IP address is the most important metric that they are going to check, all the views coming from the same IP address seems to be fake.

Figure 6 shows the number of views per day counted by YouTube in the YouTube Analytics of these robots in the first six days. Notice that we cannot compare the number of views of the first and the last days, as it depends on the hour at which we started and stopped the video.

As expected, most of the views of *No proxies* bot do not count, meaning that the IP address is an important factor. One detail is that YouTube did not remove this video, even when all the views come from the same IP address. The *View duration = 0* bot obtains a number of views similar to the *Normal* bot most of the days. This could mean that YouTube does not check the view duration for counting one view as valid. Comparing the views of the *Normal* bot and the *Only proxies* bot, we see that the second one gets more views than the first one. Thus, when we remove modules and make the robot simpler, YouTube counts more views per day. We do not know why this could happen.

In all these cases, our robot was making views that YouTube was not counting. In the next sections we are going to try to see why this could happen.

D. Ratio of counted views

As we saw in all the bots that we run, YouTube is not counting all the views. For this reason we decided to create a log in which the robot would annotate views done successfully as soon as it starts playing the video. In this log it annotates the timestamp, the proxy used, the traffic source, if it is a signed user and if the cookies were recovered or not.

The following robots were configured:

- Normal: as in the previous cases it is a robot with all the modules configured.
- No recovered cookies: a bot that does not recover the cookies previously configured. Thus, for YouTube, every view is a new user. It has all the modules configured except Signed in users.
- Deterministic: the purpose of this robot is that there is no difference between views. Thus, all the views of this

robots have the same user-agent and view duration, it does not recover cookies, it does not go to related videos and it does not use signed users. Thus, all the views are the same, the only factor that changes is the proxy (as we saw the IP address is an important factor).
- Deterministic + signed in users: same as the previous bot, but it has signed user that click like or dislike.
- Deterministic 25 proxies: same robot as the deterministic but it only has 25 proxies to choose. Thus, we are reducing the number of IP addresses available in order to know if it is a factor that they check.

As we did not configured these robots with the same view interarrival time, we define the metric ration in equation 3 in order to compare them.

$$Ratio = \frac{views_{Analytics}}{views_{Log}} \quad (3)$$

With this metric we obtain the percentage the views counted by YouTube as $views_{Analytics}$ are the views counted by YouTube in Analytics, and $views_{Log}$ are the views counted by our robot in the log.

We run these robots for seven days, obtaining the ratios shown in figure 7. Our experiments show that the robots *Normal*, *No recovered cookies* and *Deterministic* have similar ratio of success. In the case of *Deterministic+signed in users* the ratio is 10% higher, showing that YouTube check system is less restrictive when signed in users watch videos. The surprising result is that the *25 Proxies* robot has a higher ratio than the *Deterministic* robot. As commented before, the difference of these robots is the number of proxies from which they can connect. The robot with less number of proxies obtains higher ratio, meaning that the IP address is a important factor but not decisive.

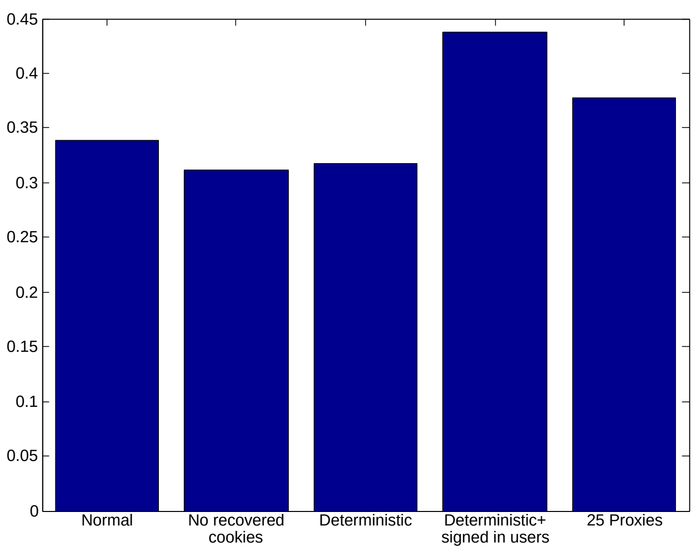


Fig. 7. Ratio of views

E. Comparison against "real users": Webpage results

We asked ourselves the following questions: is YouTube removing views because they detect that it is a robot? Or does it happen to all the videos? The main problem to answer these questions is that only YouTube registers all the views that a video received from all traffic sources.

However, we can create a webpage to watch the video embedded on it and register the users that play the video. Then, we can compare these data with the views in Analytics coming from that webpage, that is, having our webpage as a playback location.

Experiment

We use YouTube Player API Reference for iframe Embeds[6] to retrieve the state of the player (unstarted, ended, playing, paused and buffering). We captured the starting of the video (when the video changes its state from unstarted to playing). When this event was captured a request was sent from the web browser to the server using AJAX. In the server side, we store the ID of the video, the timestamp, the IP address and the user-agent.

We wanted to have a considerable number of views, at least more than 301, so we asked 1000 people to watched the video once using a crowdsourcing website.

Results

In table II we present the results of the experiment. As we can see we capture 990 views from 929 different IP addresses in our webpage, meanwhile YouTube only counted 572 views coming from it.

	Webpage	YouTube	Ratio
Views	990	572	57.77%

TABLE II
SUMMARY OF THE EXPERIMENT

This means that even when views are real, YouTube checker is removing them for unknown reason.

VI. COMPARISON AGAINST OTHER APPROACHES: THE BUSINESS OF VIEWS

Fiverr[7] is a crowdsourcing webpage in which people offer to do a task for 5$, for example preparing a logo for a business, translating texts, programming a piece of code, etc. But there are other tasks like doing fake likes in Facebook, increasing the number of followers in Twitter, retweeting a tweet or increasing the number of views in a YouTube video. There are people that make 5.000, 10.000 or even 100.000 views in a video in a few hours. Besides some of them offer likes, comments and subscribers. We hired three of these offers finding interesting insights.

One of these offers advertised 20.000 views without using any bot and all the views are embedded in Facebook. They got more than 20.000 views in 8 hours. Looking at the Analytics provided by YouTube we see that, as promised, all the views were coming from Facebook. The views are coming from three different locations (figure 8): Netherlands, United States and Germany. The surprising metric is that the average view

[6]https://developers.google.com/youtube/iframe_ api_ reference
[7]http://www.fiverr.com/

duration is 0 seconds and the estimated minutes watched 0 minutes. This means that their system does not watch the video and that YouTube counts those views as valid . Besides 100% of the views have demographics data.

GEOGRAPHY	VIEWS ↓	ESTIMATED MINUTES WATCHED	AVERAGE VIEW DURATIO N
Netherlands	12,106 (48.3%)	0 (0.0%)	0:00
United States	7,531 (30.0%)	0 (0.0%)	0:00
Germany	5,423 (21.6%)	0 (0.0%)	0:00
Spain	3 (0.0%)	2 (100.0%)	0:32

1–4 of 4

Fig. 8. YouTube Analytics of one of the

Another offer advertised 20.000 views, 30 likes 40 subscribers and 30 favorites. This system overcame the 301 limit in 12 hours and daily, it was increasing the viewcounter in 4.000 views on average. Checking the Analytics we see that the views came from 105 different locations. We found two different playback locations: Mobile device and embedded in Facebook. The views from Mobile devices were coming with a direct link and they had an average view duration of 0 seconds and estimated minutes watched of 0 like the previous system.

The last offer advertised 1000 views with a natural pattern, saying that they have 140 people watching videos. This system is slower than the previous two, as they were getting an average of 200 views per day. They used 103 different locations. In this case the average view duration and the estimated minutes watched seem to be normal. The traffic sources were Twitter, Facebook and suggested videos. When we checked the suggested videos of those ones that appeared in the statistics, we did not see our video. So, it could be that the requests were implemented sending a referrer field in the HTTP header. However, this system seems to be the most natural.

These examples demonstrate that YouTube views check system is primitive and people can overcome the 301 views limit easily, making business with YouTube weaknesses. Many of these systems would not work if YouTube had a different check system in which they verified all the views using metrics that they have, such us the view duration. There are other systems, like the last offer that we hired or our robot, that it is more difficult to say if the views are real or not, but these systems are not able to do views as fast as the two previous ones.

VII. Conclusions and Future Work

In this paper, we have conducted an experimental analysis to gain insights on the check system for counting views implemented on YouTube. In particular, one of the key goals of our analysis was to analyze if it is possible to overcome the 301 view limit of YouTube at which YouTube checks if views are real. For this purpose, we implemented a robot that makes fake views in YouTube.

Our aim was to develop a robot with a behavior as natural and realistic as possible. To do this, we crawled some videos and analyzed the evolution of some statistics such as views, likes and dislikes. Additionally, we also analyzed the main traffic sources and the percentage of each of them. All this behavior was replicted in our robot.

A number of lessons have been learnt from this analysis. One of the key results of the analysis is that overcoming the 301 view limit is easy to do. We also found out that YouTube does not check all the HTTP headers.

Another interesting conclusion of the experimental evaluation conducted is that even for a robot that closely follows real users' behavior, YouTube does not counting all the views performed. In order to gain insight into this, so conducted an experiment with real users watching a video: results show that even if with such real uers, YouTube is not counting all of views.

In order to compare the behavior of our robot against other approaches to make fake views, we evaluated the performance of some of the fake views, which are offered in the Internet at a certain price. We hired some of these services and we found out that they take advantage of some of the YouTube weaknesses, such as the fact that the view duration is not taken into account. If YouTube had a more sophisticated check system, such fake views would not go undetected.

As future work, we plan to work on the following two directions:

- We plan to extend the present analysis in order to confirm the reason why YouTube does not count all the views even with real users. To this end, we plan to look more closely at the impact of the number of IP addresses used as well as to devise additional experiments with real users.
- We plan to propose novel algorithms for detecting fake vies that outperforms the current system used by YouTube, which has been shown by our experiments to perform poorly.

References

[1] X. Cheng, C. Dale, and J. Liu, "Statistics and social network of youtube videos," in Quality of Service, 2008. IWQoS 2008. 16th International Workshop on, pp. 229–238, 2008.

[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, IMC '07, (New York, NY, USA), pp. 1–14, ACM, 2007.

[3] G. Szabó and B. A. Huberman, "Predicting the popularity of online content," CoRR, vol. abs/0811.0405, 2008.

[4] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini, "A peek into the future: predicting the evolution of popularity in user generated content," in Proceedings of the sixth ACM international conference on Web search and data mining, WSDM '13, (New York, NY, USA), pp. 607–616, ACM, 2013.

[5] G. Grsun, M. Crovella, and I. Matta, "Describing and forecasting video access patterns.," in INFOCOM, pp. 16–20, IEEE, 2011.

[6] H. Pinto, J. M. Almeida, and M. A. Gonçalves, "Using early view patterns to predict the popularity of youtube videos," in Proceedings of the sixth ACM international conference on Web search and data mining, WSDM '13, (New York, NY, USA), pp. 365–374, ACM, 2013.

[7] F. Figueiredo, F. Benevenuto, and J. M. Almeida, "The tube over time: characterizing popularity growth of youtube videos," in Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11, (New York, NY, USA), pp. 745–754, ACM, 2011.

[8] R. Zhou, S. Khemmarat, and L. Gao, "The impact of youtube recommendation system on video views," in Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, IMC '10, (New York, NY, USA), pp. 404–410, ACM, 2010.

[9] Y. Borghol, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti, "The untold story of the clones: content-agnostic factors that impact youtube video popularity," in Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '12, (New York, NY, USA), pp. 1186–1194, ACM, 2012.

[10] A. Brodersen, S. Scellato, and M. Wattenhofer, "Youtube around the world: geographic popularity of videos," in Proceedings of the 21st international conference on World Wide Web, WWW '12, (New York, NY, USA), pp. 241–250, ACM, 2012.

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

[12] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of youtube network traffic at a campus network - measurements, models, and implications," Comput. Netw., vol. 53, pp. 501–514, Mar. 2009.

[13] R. Crane and D. Sornette, "Robust dynamic classes revealed by measuring the response function of a social system," Proceedings of the National Academy of Sciences, vol. 105, no. 41, pp. 15649–15653, 2008.