enhanced COntent distribUtion with Social INformation
www.ict-ecousin.eu

Deliverable D3.3

Measurement, Modelling & Prediction of Social-Content Interdependencies (Final Version)

Public deliverable, Version 1.0, 16 December 2014

Authors	
Orange	Ali Gouta
AL-BELL	Danny De Vleeschauwer, Chris Hawinkel
IMDEA	Miriam Marciel
TSP	Ángel Cuevas, Reza Farahbakhsh, Noel Crespi
TUD	Julius Rückert, Tamara Knierim, Christian Koch, Leonhard Nobach, David Hausheer (Editor)
TI	Fabio Luciano Mondin
UC3M	Ruben Cuevas, Juan Miguel Carrascosa
Reviewer	Yannick Le Louedec

Abstract

Following deliverable D3.1, the aim of this deliverable D3.3 is to specify the final mechanisms for measurement, modelling, and prediction of social-content interdependencies. As such, D3.3 outlines the final design of active large-scale crawling tools and passive measurement mechanisms to collect content-related information in online social networks (OSNs) and content distribution infrastructures and applications. Moreover, D3.3 provides the final approaches on modelling of different aspects related to social network users and content, e.g., user behaviour, social network graphs, and content popularity. Finally, D3.3 provides the final design of mechanisms for prediction of content popularity and social cascades of contents as well as completed analyses of the impact of the social dimension on content interests.

Project funded by the European Union under the
Information and Communication Technologies FP7 Cooperation Programme
Grant Agreement number 318398

EXECUTIVE SUMMARY

Following the initial version documented in deliverable D3.1 [D3.1], the aim of this deliverable D3.3 is to specify the final mechanisms for measurement, modelling, and prediction of social-content interdependencies. Towards this end, D3.3 outlines the final design of active large-scale crawling tools and passive measurement mechanisms to collect content-related information in Online Social Networks (OSNs) and content distribution infrastructures and applications (e.g., BitTorrent, YouTube, BTLive). Moreover, D3.3 provides the final approaches on modelling of different aspects related to social network users and content, e.g., user behaviour, social network graphs, and content popularity. Finally, D3.3 provides the final design of mechanisms for prediction of content popularity and social cascades of contents as well as initial analyses of the impact of the social dimension on content interests.

Therefore, the main results of this deliverable include:

- An update of active crawling and passive measurement mechanisms (Section 1). This includes a documentation of the social observer / social data collector implemented by Telecom Italia.
- A description of social interaction models (Section 2). This includes a prediction of interest similarity based on Facebook, a prediction of user location using partial information available in Facebook profiles, as well as a characterization of professional publisher activity across Twitter, Facebook and Google+. Furthermore, an analysis of BTLive content distribution, an analysis of partial and social aware prefetching in YouTube, as well as an analysis of fake views in Youtube are provided here. The section is completed with a description of exploring D2D opportunity using the Orange traces and an analysis of Orange's CUTV dataset, TSP's Facebook data, and the Flixter dataset.
- An update on social-aware content diffusion approaches (Section 3). The outcome of this work includes a microscopic information propagation analysis in Google+ and its comparison with Twitter, as well as macroscopic geographical information propagation analysis in Twitter.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

On-line Social Networks (OSNs) have become the most successful application in the Internet in only few years accounting with billions of users every day. The huge success of these systems along with the enormous possibilities they open to end users and commercial players have defined their study as a hot topic in the research arena. Therefore, the study of OSNs and the way in which they can be exploited by third parties is a core element of the eCOUSIN project. A necessary first step was to gather data from various OSNs to later analyse them and derive social interactions required for the development of the project. For this purpose, the eCOUSIN consortium – specifically WP3 – was developing tools to gather data from different OSNs such as Twitter, Facebook and Google+.

Correspondingly, eCOUSIN WP3 includes three tasks: Task 3.1 dealt with the design of active crawling and passive *measurement* tools to collect content-related information in Online Social Networks (OSNs) and content distribution infrastructures and applications. Furthermore, Task 3.2 developed approaches on *modelling* of different aspects related to social network users and content, e.g., user behaviour, social network graphs, and content popularity. Finally, Task 3.3 designed mechanisms for *prediction* of content popularity and social cascades of contents as well as initial analyses of the impact of the social dimension on content interests.

Thus, following the initial version documented in deliverable D3.1, the aim of this deliverable D3.3 is to provide an update about those measurement tools and datasets that have been collected in the scope of WP3. Furthermore, the final results on modelling and prediction of social-content interdependencies are presented.

Specific activities related to Task 3.1 include the design and implementation of efficient algorithms for crawling, monitoring, and data gathering in social-based content centric infrastructures and large scale crawling and measurement of OSNs as well as measurement of content distribution and content portals and monitoring of OSN and content distribution traffic in operational networks. Correspondingly, Section 1 provides an update on active crawling and passive measurement mechanisms. This includes a documentation of the social observer / social data collector implemented by Telecom Italia.

Task 3.2 specifically developed models that allow describing user behaviours and user incentives in terms of social networks and content distribution. Furthermore, it models user influence in content popularity, online social networks usage patterns, and content dynamics, as well as identifies and analyses social relationships, social network structures, and social interaction models. Correspondingly, Section 2 describes social interaction models. This includes a prediction of interest similarity based on Facebook, a prediction of user location using partial information available in Facebook profiles, as well as a characterization of professional publisher activity across Twitter, Facebook and Google+. Furthermore, an analysis of BTLive content distribution, an analysis of partial and social aware prefetching in YouTube, as well as an analysis of fake views in Youtube are provided here. The section is completed with a description of exploring D2D opportunity using the Orange traces and an analysis of Orange's CUTV dataset, TSP's Facebook data, and the Flixter dataset.

Moreover, Task 3.3 specifically dealt with the prediction of popularity of content and content consumption patterns driven by social networks, prediction of social cascades, and the analysis of the impact of the social dimension on content interests. Correspondingly, Section 3 provides an update on social-aware content diffusion approaches. The outcome of this work includes a microscopic information propagation analysis in Google+ and its comparison with Twitter, as well as macroscopic geographical information propagation analysis in Twitter.

The deliverable is completed by Section 5 which summarizes the main findings and provides final conclusions.

1. ACTIVE CRAWLING AND PASSIVE MEASUREMENTS

The aim of the first task (Task 3.1) of WP3 was to design large scale monitoring tools based on active crawling and passive measurement mechanisms for obtaining content information related to online social networks (OSNs) and content distribution infrastructures. The tools developed in this task include real-time monitoring mechanisms to collect publicly available information from distributed vantage points (e.g., using Planetlab) as well as to implement tools to analyse network traffic in local environments (e.g. a university campus) and in an operator's network, using dedicated probes. These mechanisms were specifically tailored at different OSNs (e.g., Facebook, Twitter, and Google+), content distribution networks (e.g., BitTorrent), and large content portals (e.g., YouTube). The design of monitoring tools was based on tree- or mesh/gossip-based mechanisms with the goal to collect accurate data from these infrastructures in a scalable and robust manner and at a low overhead.

A detailed overview on crawling tools developed in the consortium was already given in [D3.1]. Accordingly, only a minor update of active crawling and passive measurement mechanisms is presented here. This includes a documentation of the social observer / social data collector implemented by Telecom Italia.

1.1 Social Observer / Social Data Collector

Social Observer is the name of the implementation of the Social Data Collector Module into the eCOUSIN architecture for the Personal Sharing Clouds Use Case.

Figure 1 and the associated explanations are extracted from Deliverable D2.1. They summarize the different functional steps of this use case:

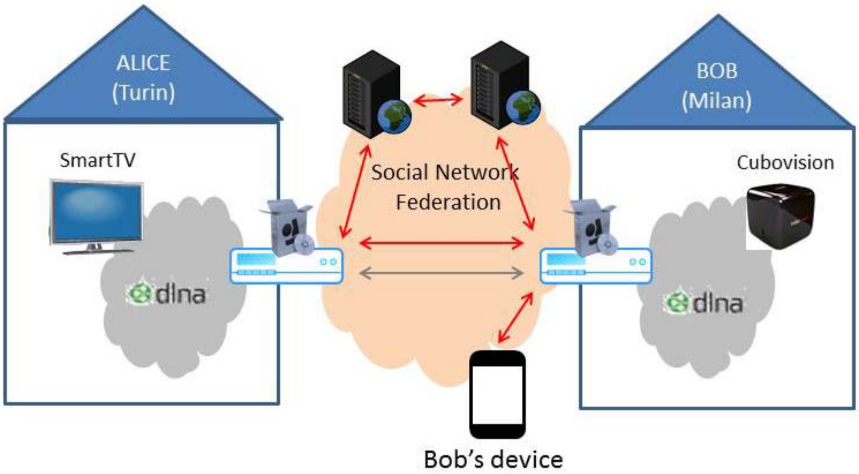


Figure 1. Personal content sharing clouds

A. Alice has a DLNA-enabled Smart TV she usually exploits to browse the Internet, e.g., to read news and access Facebook. Bob has a Telecom Italia's Cubovision (or any other similar set-top box) where he stores his own pictures and music files, to be available on other remote nodes via DLNA AV Media Server. Smart TV and Telecom Italia's Cubovision devices are deployed in Alice and Bob home networks respectively;

B. Alice and Bob meet on their summer holiday; when they come back home, they become friends on their preferred Online Social Networks (OSNs), eventually exploiting the Federated Social Networking Standard Implementations (FSN). The creation of a new social relationship triggers the automatic generation of a communication link among Alice's and Bob's gateways, already deployed in their home networks to provide Internet connectivity;

C. Once friends, Alice uses her Smart TV to access via DLNA the pictures, videos and music that Bob stores on his Telecom Italia's Cubovision and has tagged as "shared with friends". Alice accesses Bob's Telecom Italia's Cubovision without any specific goal, just for her own curiosity to see pictures and videos of her new friend, e.g., to view photos of his previous vacations or check his music selection;

D. After a couple of days, Bob saves his holiday pictures in the "Summer 2012" directory on the Telecom Italia's Cubovision and then notifies via FSN his friends (including Alice) of the new photo album. Alice clicks on the FSN post exploiting her Smart TV and accesses via HTTP the whole directory content;

E. Bob takes a new video with his smartphone. The FSN client automatically uploads the video on his Telecom Italia's Cubovision at home and notifies his friends of the new video location. Alice can access that specific video stored in the Bob's home private network simply clicking on the new post.

F. Finally, Bob accesses a YouTube video from his mobile and decides to share it in using one or more OSNs. Automatically, this video is downloaded and stored in his Telecom Italia's Cubovision system. While Alice is browsing her OSNs contacts she clicks on the YouTube video Bob shared, and the Content provider (i.e. YouTube) redirects Alice to Bob's Cubovision that serves the video to Alice.

The technical architecture implementing this use case is depicted on Figure 2 and the modules composing it are detailed below, with a specific focus on the Social Observer.

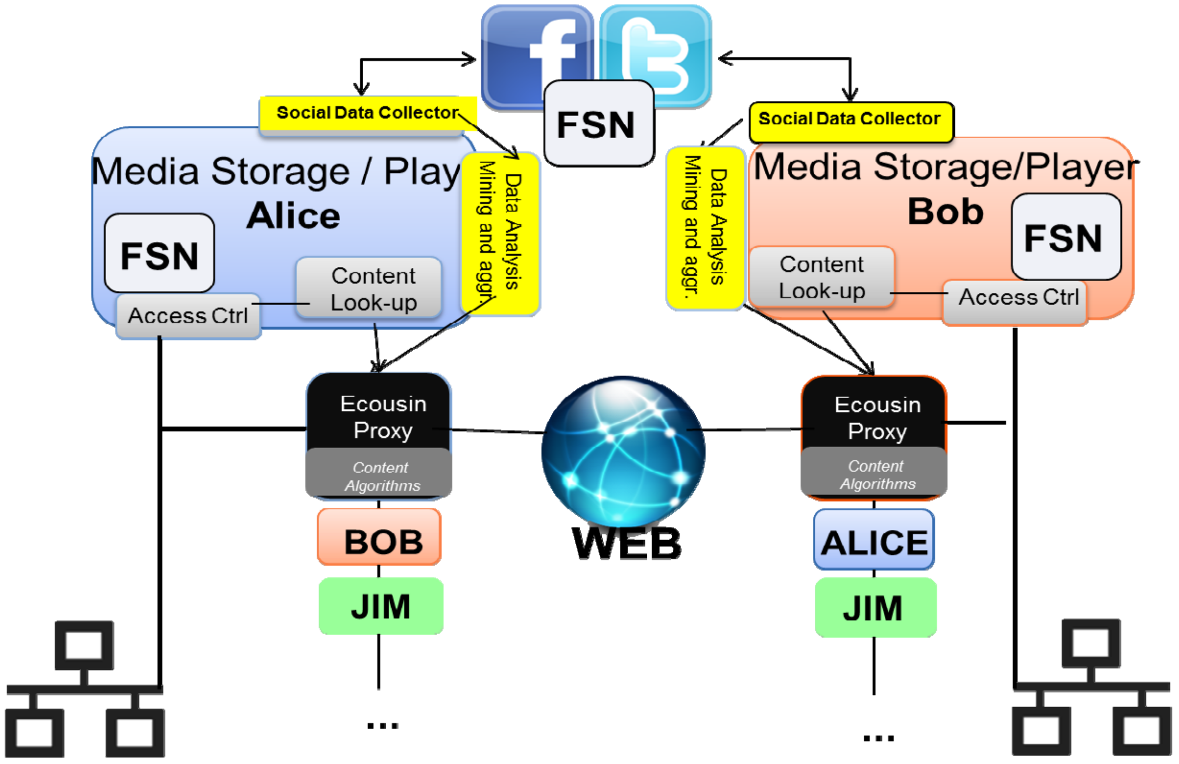


Figure 2. Personal content sharing clouds detailed architecture

The social data collector shown on Figure 2 is responsible for interfacing online social networks with the media centres. It provides the following functionality:

- Allow users to connect their social network identities to the media centres. For example, in the case where Facebook is the considered OSN, the user links his Facebook account to the media centre using the Facebook Connect API.
- Explore the user's social relationships in order to detect the OSN friends/followers who have already an account linked to a media centre ;
- Collect all data required to set direct connections among media centres: IP address of the media center (if it has more than one interface, the one hosting the service), TCP port, status (online/offline);
- Monitor changes in the user's social relationships as well as changes in the parameters used for connecting media centres;
- (Optionally) Publish information about the possibility to connect to media centres. Since this module uses social networks as meeting point, it gives the possibility to publish on the social network the info above (IP address, port and eventually some extra meta-data about the type of hardware). For Facebook, for example, the media centre leverages on a Facebook APP, which is responsible both to scan friends to grab connection data and publish a user's connection data on his wall if he wants.

From the user's viewpoint, he connects to the personal content sharing cloud service via a specific App. He gives the right to the service to access to his list of OSN friends and the right to publish information related to his media centre. Then, the social data collector may access to his list of friends of this newly connected user and can detect the ones who have also subscribed to the service. The social data collector saves locally the information on how to access their media centre and it passes this information to the other modules, which are responsible for setting the network connection and for locating shared resources and content.

During the prototype implementation phase (reported in WP6) we came up against some issues with the users who have a high number of social relationships. We plan to fix this issues for the final prototype implementation. The app to connect to the system will also be enriched in the final prototype to ease the process for the user. Both modifications will involve the Social Observer.

The App Mechanism is a typical Facebook Mechanism to collect and exchange information among users, but in general any mechanism to collect information about who the user wants to reach and how to reach him/her is something that can be managed into this module as it is for example in the case of Federated Social Networks.

The Opensocial/Ostatus Protocol Suite aims at standardizing the interactions among different social networks, in order to overtake the typical "walled garden" paradigm. The basic idea is to have an account on a preferred social network, which is able to give access to some content (matching specific rules) to users belonging to other social networks.

This last feature could open the way to a scenario in which the "social network" is actually a "home social network", thanks to a software running on the media centre itself. The Opensocial/Ostatus Protocol suite, which is now becoming a W3C standard, covers both client to server and server to server communication. So we envisioned a two-fold level of integration with the personal sharing clouds use case, both involving the Social Observer:

- Exploit Opensocial to publish data (for example via Mobile) on the media center: In this case the files published standing on specific folder into the media centre will automatically get accessible as local resources from remote endpoints according to the normal behaviour of the use case ;
- Exploit Opensocial to collect information on how to connect to my friends' remote media centre searching for this information into the profile data.

2. SOCIAL INTERACTIONS MODELLING

In the second task of WP3 (Task 3.2), the data collected by the tools developed in Task 3.1 (as reported in Section 1 above as well as [D3.1]) were used to study, analyse, and model different aspects related to social network users and content dynamics in local domains. These aspects include but were not limited to user incentives that influence user behaviour, content popularity, online social networks usage patterns (e.g., rate of tweets posted in twitter), as well as user's social network structure based on social relationships between users. Those aspects were used to model the links between OSNs and the underlying content distribution networks and to infer key social properties, such as ad hoc communities in large social networks, that drive the content distribution/information cascade and that could be exploited to enhance content distribution networks, e.g., by using them as input for content placement.

An initial set of models for user behaviour, content popularity, and online social networks usage patterns was documented in [D3.1]. Accordingly, this section provides a description of the final social interaction models. This includes a prediction of interest similarity based on Facebook, a prediction of user location using partial information available in Facebook profiles, as well as a characterization of professional publisher activity across Twitter, Facebook and Google+. Furthermore, an analysis of BTLive content distribution, an analysis of partial and social aware prefetching in YouTube, as well as an analysis of fake views in Youtube are provided here. The section is completed with a description of exploring D2D opportunity using the Orange traces and an analysis of Orange's CUTV dataset, TSP's Facebook data, and the Flixter dataset.

2.1 Prediction of Interest Similarity based on FB

Online Social Networks (OSNs) have boomed and attracted a huge number of people to join them over the last decade. In OSNs, the participants publish their profiles, make friends, and produce various content (photos, answers/questions, videos, etc.). Unlike legacy web systems, OSNs are organized around both people and content, which provide us with unprecedented opportunities to understand human relationships, human communities, human behaviours and human preferences [MMG07] [LH08] [UKB11].

With the evolution of OSNs, understanding to which extent two individuals are alike in their interests (i.e., interest similarity) has become a basic requirement for the organization and maintenance of vibrant OSNs. On the one hand, users' interest similarity could be leveraged to support friend recommendation and social circle maintenance. For instance, the decision to recommend the users who share many interests with each other to be friends could increase users' approval rate of recommendation, because people usually aggregate by their mutual interests [LGK12]. On the other hand, knowing interest similarity between users also facilitates social applications and advertising. For example, instead of randomly hunting clients, exploring those users of a high interest similarity with the existing clients could efficiently enlarge client groups for application providers and businesses.

Although many previous studies have been widely conducted on various OSN platforms, most of them have only focused on discovering various structural properties including the small world effect, community structures, and clustering [[LH08] [UKB11] [LGK12]. Such investigations could not be directly applied to the above-mentioned applications (e.g., personalized advertisements). Aiming to enhance specific social-based services and applications (e.g., friend prediction, recommendation), several existing researches have already examined how interest similarity changes with very limited social features: [AA03] [LB10] have studied that friends share more interests than strangers; [ZG07] has verified that interest similarity strongly correlates to the trust between users. However, none of them has extended this study to discuss how interest similarity varies with various social features.

Therefore, in this research, we are motivated to carry out empirical studies on how users' interest similarity relates to various social features in a wide variety of cases. In particular, we quantify interest similarity over an aggregation of user pairs based on a cosine method to capture interest overlaps between two users. Besides, we extract the social features (e.g. profile similarity, geographic distance, friend similarity) from users' social information regarding three aspects: demographic information (e.g., age, gender, location), social relations (i.e., friendship), and users' interests. Specifically, we conduct the study in three interest domains, namely movie, music and TV, over a large dataset including 479,048 users and 5,263,351 user-generated interests crawled from Facebook.

To highlight our key findings, we reveal the homophily regarding interest similarity in Facebook based on the comprehensive analysis. Generally, homophily shows homogeneity in people's social networks regarding many sociodemographic, behavioral and intrapersonal characteristics [MSC01]. Specifically, in this research:

- homophily reveals that people are more likely to be interested in the same movie, music and TV series when they are more similar in their demographic information, such as age, gender and location ;
- homophily also implies that friends have higher interest similarity than strangers do. Furthermore, the interest similarity becomes higher if two users share more common friends.

This study is distinct from the existing work on interest similarity by three aspects. Firstly, we carry out a more comprehensive analysis on the correlations between users' interest similarity and diverse social features. We attempt to dig out more relative factors which can be harnessed to enhance social recommendations and advertisement services. Secondly, the majority of existing studies on interest have not distinguished the different types of interests; they usually relied merely on users' favorite music or movies [LGK12]. Additionally, they typically measure interests in terms of genre. In this research, we consider interest similarity with respect to three interest domains: movie, music and TV, respectively. And we measure interest similarity founded on every single interest item - a finer grain.

In summary, the main contributions of this research include:

- Relying on a large dataset crawled from Facebook, the analytical results can advance the collective knowledge of OSNs ;
- The findings about homophily regarding interest similarity could practically benefit numerous applications and services, such as recommendation system and advertisement service.

2.1.1 Effects on Interest Similarity

Facebook is the largest online social network in the world, and leaves open-ended spaces for users to explicitly present their interests in several domains, such as movies, music, TV, books and so on. For studies about interest similarity among users, we crawled Facebook from March to June in 2012 and collected data from 479,048 users. To our knowledge, these data represent one of the largest and most comprehensive social information databases up to date, involving 9 interest domains and 5,263,351 user-generated interest items (including 626,294 distinct items). The analyzed data can be split into three parts: User Interest, Demographic Information, and Social Relationship:

- User Interest: We conduct the analysis across three representative interest domains - music, movie and television (TV) - since more users report interests in these three than the other domains.

- Demographic Information: It contains 7 attributes of users' profiles including age, gender, current city, hometown, high school, college and employer.
- Social Relationship: This is represented by users' friend list. The friendship relation in Facebook is bidirectional, i.e., A is B's friend when B is a friend of A. Note that we construct our dataset merely with users' public information and anonymize all the users during the analysis.

2.1.1.1 Statistical Analysis of Data

In this section, we first examine some high-level characteristics and patterns of demographics that emerge from the collective users.

1) **Demographic characteristics of individuals**: Gender, location, and age are the three specific demographic attributes being considered. 256,163 (53.5%) users in our dataset report their gender, while 173,027 (36.1%) users publish their current city which is used to represent users' location. Compared with reporting gender and current city, users are more reluctant to uncover their age and only 14, 055 (2.9%) users have their age in the crawled profiles.

	Music	Movie	TV
Male	35516	50692	40620
Female	42648	58850	47225
Male (%)	26.4	37.7	30.2
Female (%)	34.2	47.2	37.9

Table 1. Distributions of interests by gender

Among the 256,163 gender reporters, 124,677 of them are self-reported as females while 134,486 are males. Although males account for a slightly greater proportion than females in our dataset, females dominate over males of reporting interests. Table 1 presents the numbers and percentages of females and males who report their interests in terms of music, movie and TV respectively. The results infer that females are more likely to report their interests than males.

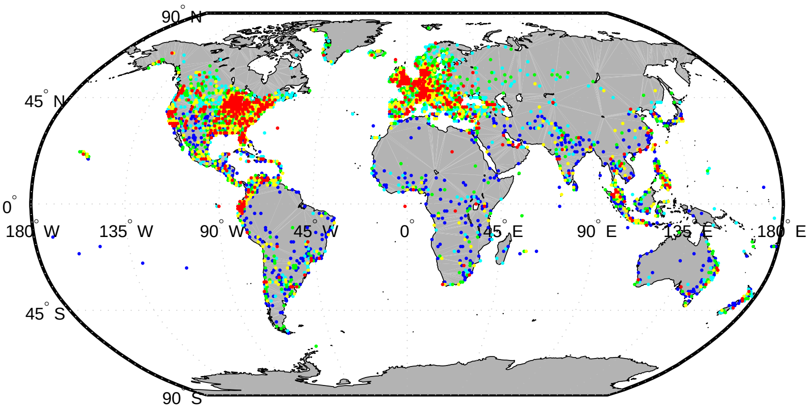


Figure 3. Location distribution

Figure 3 displays the geographical location distribution of 173,027 current city reporters over the globe. We decode the geographical coordinate of users' current city with latitude and longitude via Facebook Graph API. The color of each dot in the figure corresponds to the number of users in a city, applying a spectrum of colors ranging from blue (low), green, yellow to red (high). We can see that the red dots are mainly located in the east coast of North America as well as Europe, thus we infer that people from North America and Europe are the dominant users on Facebook. We also observe that people in coastal regions are more active than people situated inland. In addition, a few blue dots are noticed in the oceans, which might indicate some users report fake locations. We ignore them as the number is very small.

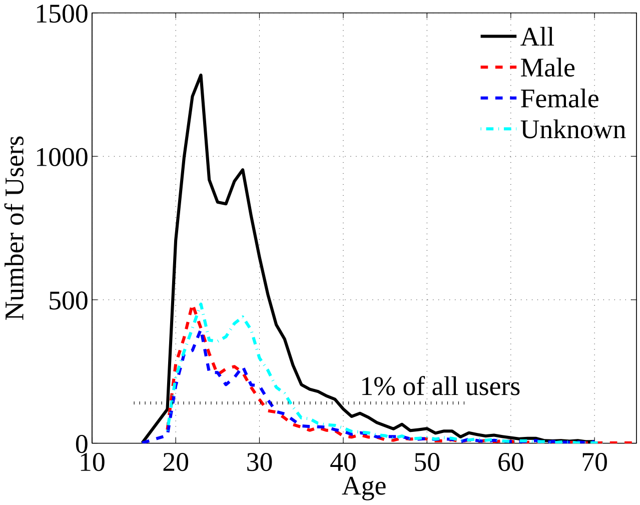


Figure 4. User distribution by age

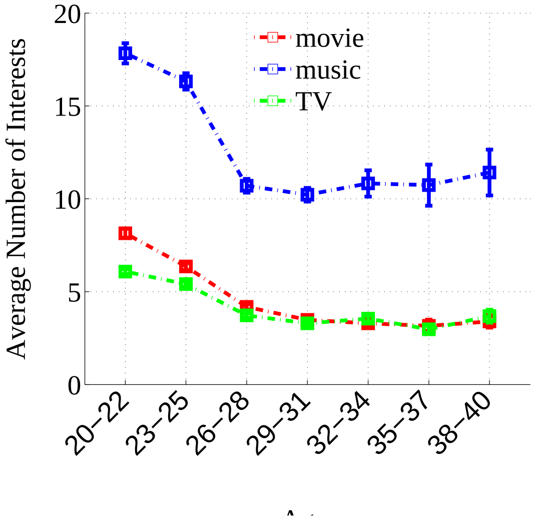


Figure 5. Interest distribution by age

Moreover, we study the distribution of users by age. Figure 4 displays the distributions of age reporters with respect to female, male, unknown gender and all. Among all the age reporters, 4196 are male and 4096 are female. We notice that the age distributions of males and females are similar to each other. We also observe that the user distributions are skewed by age following with a long tail. The users in the 20-30 span of years are the most representative users in our dataset; while the proportion of the users older than 40 or younger than 20 in our dataset is rather small (less than 10% in total). Besides, we choose 3 years as an age interval and cluster age reporters in the age range of 20-40 into seven age groups. Figure 5 examines the average number of interests that each user exhibits according to different age groups. It reveals that the young users report more interests than middle-age users.

2) Demographic characteristics of friends: In this section, we further reveal the demographic characteristics between friends in terms of gender, location, and age respectively.

We first examine the distribution of friends by gender combinations: cross-gender friends and same-gender friends. This analysis is conducted on 256,163 gender reporters. Particularly, for each gender reporter, we rely on his/her friends that are also gender reporters and calculate the percentage of friends in the same-/cross- gender respectively. Figure 6 displays the CDF of the percentage of friends by gender combinations. We observe that only around 40% of users exhibit the same gender with less than half of their friends, while more than 60% of gender reporters make fewer friends (i.e., less than half) with opposite gender. It indicates that people prefer to make friends with others of the same gender, especially for men.

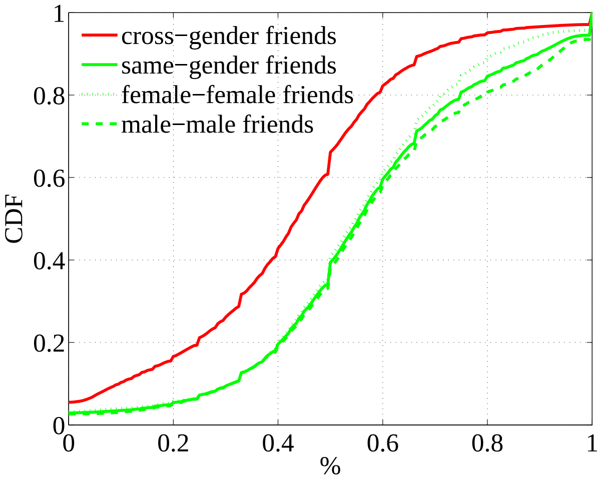


Figure 6. CDF of friends distribution by gender combination

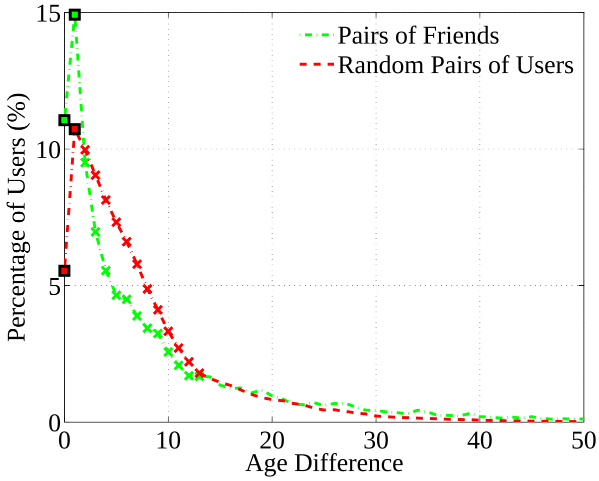


Figure 7. Pairs distribution by age difference

In addition, we track how age affects the friendship between people. Figure 7 displays the distribution of pairs at various age differences. It reveals that people are more likely to make friends with others at the same age or at an age gap of 1-2 years. The percentage of friend pairs decreases rapidly as age difference increases when it is larger than 1 year. Besides, we also notice that the percentages of friend pairs are less than the numbers of random pairs at the age differences in the range of 3 − 13 years. When age difference is larger than 13 years, people make friends following the random probabilities. We infer that people are more likely to make friends with others who are in the similar ages.

We calculate geographical distances between pairs and illustrate the pairs distribution with distances in Figure 6. From the upper subfigure, we see that the distance distribution of friend pairs is strongly skewed to the left. It falls dramatically from the start, bottoms out at the distance of 400 kilometers, and then stays at a very low value as the distance increases. Among all the friend pairs in the experiments, 28.9% of them come from the same city and 43.43% of the friends live less than 100 kilometers apart. Whereas, the lower subfigure shows that the percentages of random pairs fluctuate by distances with a gradual downward trend. The peaks and drops at some specific distances may reveal geographical characteristics. For instance, the peaks at distances of 5000 km and 6500 km may respectively indicate the width of America and the width of Atlantic. The different distributions of friend pairs and random pairs, in other words, mean that people tend to make friends within a short distance.

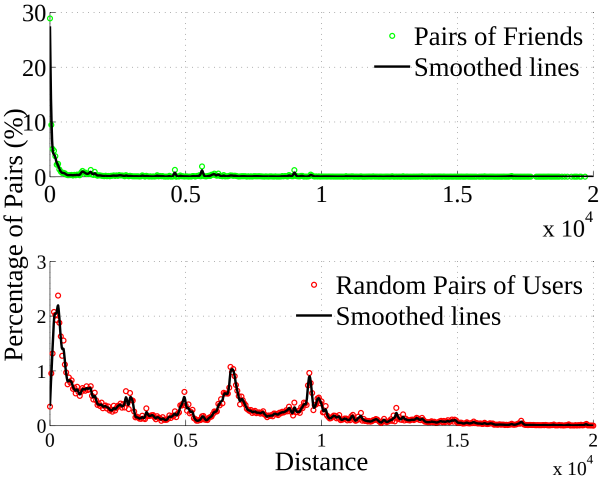


Figure 8. Pairs distribution by pairs

2.1.2 Effects on Interest Similarity

In this section, we first define the metric of interest similarity, followed by the studies on how interest similarity correlates to demographic information, social relationships and interest individuality sequentially.

2.1.2.1 Definition of Interest Similarity

We formalize a notion of Interest Similarity that measures how much two users' interests overlap. We denote user u's interests by an interest set I_u instead of a binary interest vector, in order to avoid the very sparse interest vector. Drawing on the calculation of cosine similarity, interest similarity between users u and v is then defined as the cosine distance between their respective interests sets as:

$$s_I(u,v) = \frac{||I_u \bigcap I_v||_1}{||I_u||_2 \cdot ||I_v||_2}$$

In the dataset, each user might report various items in various interest domains. We think of users' interest similarity separately in different domains, i.e., interest similarity in terms of movie, music, TV. For the analysis of each particular interest domain, we only consider the users who have more than three items in the domain.

2.1.2.2 Homophily of Interest Similarity by Demographics

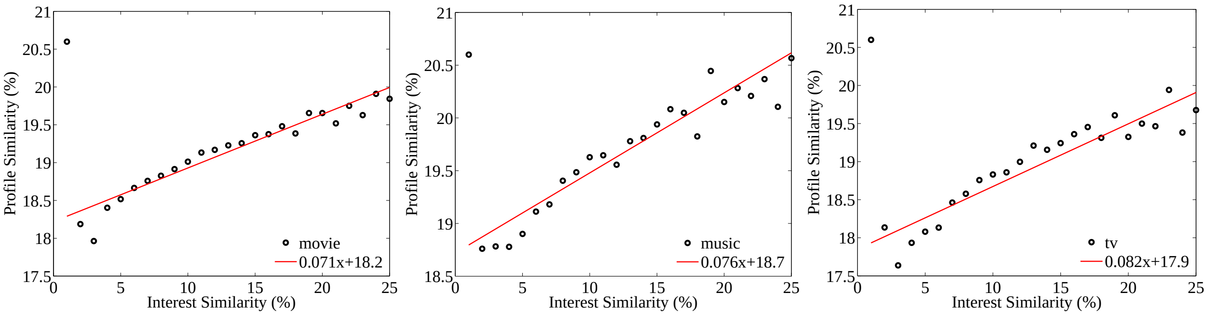
In this section, we study how demographic information affects interest similarity between users. We separately conduct several experiments by using different user samples. For instance, to test the relation between gender and interest similarity on movie, we select users who present gender and more than three interested movies and construct a gender/movie set of pairs.

1) Profile similarity with interest similarity: We first look into how the interest similarity between users changes with their profile similarity. Similar to the interest similarity evaluation, we perform cosine to profile vectors of two users and formulate profile similarity as:

$$s_p(u, v) = \frac{\|P_u \bigcap P_v\|_1}{\|P_u\|_2 \cdot \|P_v\|_2}$$

In particular, 7 demographic attributes of age, gender, current city, hometown, high school, college and employer are considered.

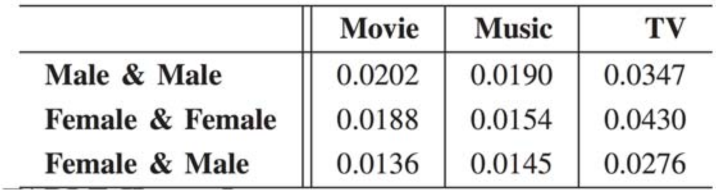
We generate 500,000 user pairs for each interest domain and show the collective relation between interest similarity and profile similarity in Figure 9. Regarding all the three interest domains of movie, music and TV, we observe that the profile similarity gets higher if the users share more common interests. We can fit their relations with linear functions as y = ax + b. In other words, the observations reveal the positive correlation between interest similarity and profile similarity regardless of interest domains, whereas the coefficients are different in these domains.



(a) Movie (b) Music (c) TV

Figure 9. Profile similarity with interest similarity

2) Interest similarity by gender: We group user pairs according to the categories of their gender combinations. Table 2 shows the average interest similarities of male-male pairs, female-female pairs, as well as male-female pairs. We observe that people have a higher interest similarity with the others when they are in the same sex. For instance, the interest similarity regarding movies between two males is close to 0.02 while the value between female and male only exhibits 0.014. This demonstrates that the homophily of interest similarity holds for gender.



	Movie	Music	TV
Male & Male	0.0202	0.0190	0.0347
Female & Female	0.0188	0.0154	0.0430
Female & Male	0.0136	0.0145	0.0276

Table 2. Interest similarity by gender

In addition, we also notice that user pairs share much higher interest similarity in terms of TV than the other two interest domains. For example, the male-female user pairs generate an average interest similarity of 0.028 regarding TV, compared with 0.014 and 0.015 for movie and music respectively. It might be due to the fewer selections for TV shows (there are 66, 396, 93, 846 and 370, 456 distinct items of TV, movie and music respectively in our dataset). Moreover, we find that males are more alike to each other on the interests of movie and music whereas females have higher similarity in the domain of TV.

3) Interest similarity by distance: We intuitively hypothesized that the pairs would exhibit a higher interest similarity if they are geographically closer to each other. Figure 10 plots the aggregate relation between distance and interest similarity based on 500,000 pairs in each interest domains. We observe that interest similarity changes with the distance between users: the average distance of pairs decreases with the increase of interest similarity.

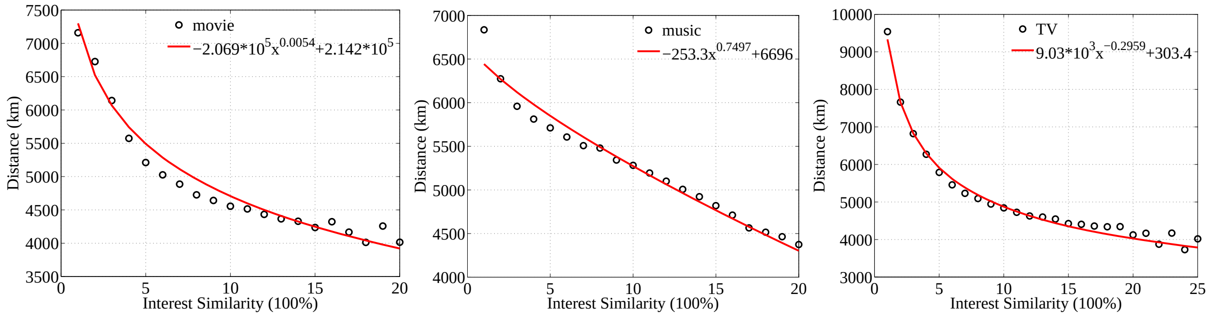
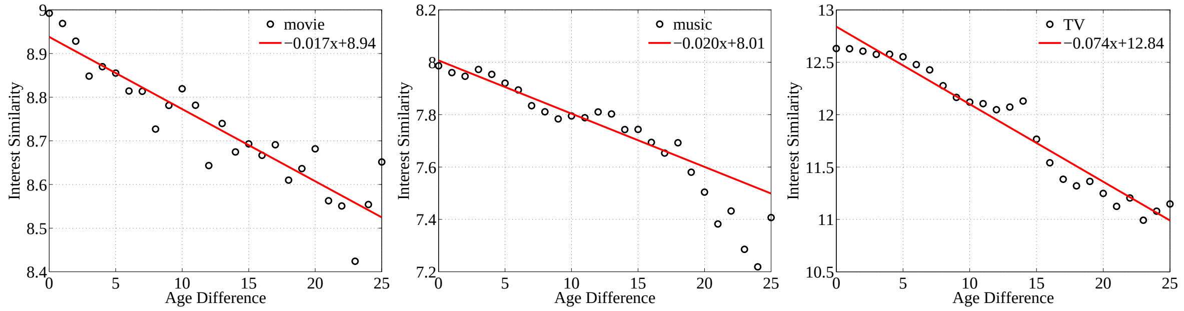


Figure 10. Geographical distance with interest similarity

4) Interest similarity by age: Intuitively, people in various generations appreciate diverse styles of music or have different tastes of movies in specific areas. For instance, young generation of 1990s probably likes Justin Bieber; while middle-age people born in 1970s might listen to the music from The Beatles more. Therefore, in this section, we are interested in how the age difference influences the interest similarity of pairs.

According to the distribution of users by age (shown in Figure 4), the experiments in this section only depend on the users whose age falls between 20 and 45 years. Therefore, the age difference ranges from 0 to 25 years. In addition, although the number of age reporters is relatively small (14,055), the amount of user pairs generated by randomly coupling users is huge enough. We also produce 500,000 user pairs for each interest domain.

Figure 11 displays how the interest similarity of user pairs changes with their age difference. We observe that the interest similarity declines as the age difference goes up with respect to all the three interest domains. This observation demonstrates that the homophily of interest similarity holds for age - the users share more interests if they are more similar at age. We employ linear models to depict the trends of the correlations.

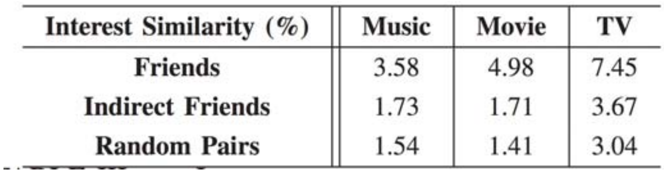


(a) Movie (b) Music (c) TV

Figure 11. Age difference with interest similarity

2.1.2.3 Friendliness of Interest Similarity

Relationship is considered as a special element, which distinguishes social network from general web sites and blogs. With user-generated relationships, OSNs are constructed by connecting people. These relationships generally involve many real social relations. For example, the friends on OSNs perhaps have known each other in their real life or have engaged in a same event or in a same interest group. We hypothesize that the friendship among users in Facebook would strongly correlates to their interest similarity. And the examinations are carried out in two parts: the effects of friendship relations of pairs and quantified friend similarity.



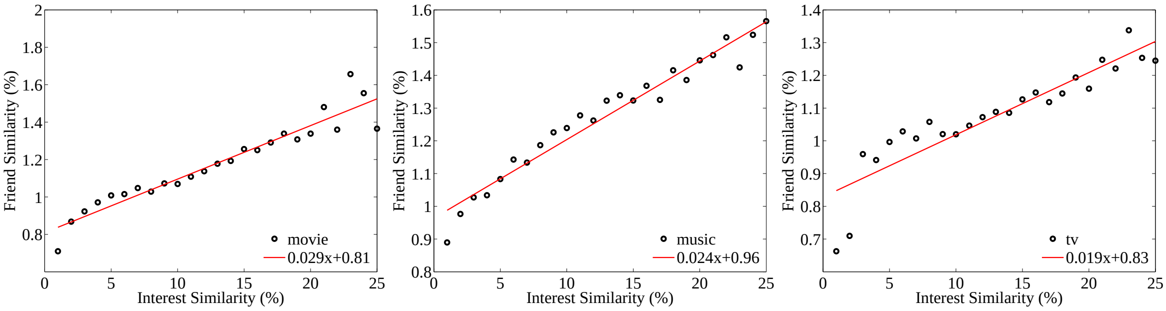
Interest Similarity (%)	Music	Movie	TV
Friends	3.58	4.98	7.45
Indirect Friends	1.73	1.71	3.67
Random Pairs	1.54	1.41	3.04

Table 3. Interest similarity by friendship

1) Interest similarity by relation of pairs: We take into account users' friendships by two hops and categorize users' relations into three groups: pairs of friends, pairs of indirect friends and random pairs. We define two users u and v as indirect friends if u is a friend of v's friend. We report interest similarity by friendship in Table 3. We observe that the interest similarity between friends is the highest, and indirect friends also share more interests than random pairs. For various interest domains, the average interest similarity of friend pairs could be 1 to 4 times larger than the one of random pairs. Therefore, we conclude that friends are more likely to have same tastes on any interest domain.

2) Interest similarity with friend similarity: Much previous work differentiates the relationship between two users by its strength [XNR10] [MVG10]: strong connections (e.g., intimate friends, or close friends) and weak connections (e.g., acquaintances, or strangers). In this section, we further measure the effect of relationship on interest similarity by its strength. We quantify the strength of the connection between two users by friend similarity and assume that the pairs with a stronger relationship have a higher friend similarity.

Similar to the calculation of interest similarity explained above, friend similarity is measured by the overlap of friends between two users, based on the method of cosine similarity.



(a) Movie (b) Music (c) TV

Figure 12. Friend similarity with interest similarity.

Figure 12 plots the aggregated relation of interest similarity versus friend similarity among 1,000,000 pairs for each interest domain. We observe that interest similarity is positively correlated to friend similarity in all the three interest domains. In other words, the observations demonstrate that user pairs generally share more interests if they obtain a higher friend similarity.

2.1.3 Conclusion

In this research, we conduct a comprehensive empirical study on how users' interest similarity relates to various social features in a large Facebook dataset including 479,048 users and 5,263,351 user-generated interests. We conduct the study in three interest domains (i.e. movie, music, and TV). The result reveals that interest similarity follows the homophily principle and correlates with many social features: people tend to exhibit more similar tastes if they have similar demographic information (e.g., age, location) or share more common friends. We believe the observations could be harnessed to improve various social applications and services.

2.2 Prediction of User Location using partial information available in FB profiles

During the last decade, Online Social Networks (OSNs) have successfully attracted people to share huge amount of personal information through the Internet, such as personal background, preferences and social connections. Owing to the increase of potential dangers such as stalking, identity theft and victimization in OSNs, in recent years, more and more users start to get concerned about their privacy in OSNs and become reluctant to expose all their personal information to public [RJR12]. Consequently, the users may hide some privacy-sensitive attributes (e.g., location, age, or contact information) from strangers and merely allow such information visible to their friends.

While hiding the privacy-sensitive attributes, users usually expose to public some other information that seems not private to them. As reported, the users on Facebook reveal four attributes on average and 63% of them uncover their friends list [FHC13]. Due to the correlations among various attributes, some of these self-exposed information may indicate the invisible privacy-sensitive attributes to some extent [CHC13][PMV12]. In such a case, whether the privacy-sensitive attribute that a user intends to hold is really undercover is in doubt.

In this work, leveraging users' information about location as a representative, we attempt to understand what is the risk that a user's invisible information would be exposed. Several reasons lead us to conduct this study based on location information. First, among various kinds of information, location is usually one of the privacy-sensitive attributes to a user. In real-life OSNs, we notice that users are quite careful to reveal their location information: 16% of users in Twitter reveal home city [LWC12] and 0.6% of Facebook users publish home address [BSM10]. Moreover, to third-parties, location is an interesting attribute to be utilized for commercial purposes. This may lead the third parties to discover users' hidden location information. Even worse, the discovered location information might be misused by unscrupulous businesses to bombard a user with unsolicited marketing, or even lead to more severe harms such as stalking and physical attacks [DK06]. Therefore, protecting the hidden location information for a user becomes rather critical. In particular, as Facebook is the largest and most important OSN [STE14], we concentrate on the attribute of current city in Facebook and investigate the following issues:

1) Is the private current city that a user expects to hide really undercover? In other words, if a user hides his current city but exposes some other information, can we predict a user's current city by using his self-exposed information?

2) For an individual user, can we help him understand the actual risk (probability) that his private current city could be correctly predicted based on his self-exposed information? Furthermore, can we provide some countermeasures to increase the security of his hidden current city?

To address the aforementioned issues, first, a current city prediction approach is required to predict users' hidden current city. Although many location prediction approaches have been developed for Twitter [CKM11][CCL12][IVR13][RM14] and Foursquare [PMV12][PVA12], they cannot be appropriately leveraged to Facebook because of the different properties (e.g., obtainable information) in various OSNs. For Facebook, Backstrom et al. predict users' locations based on their friends' locations [BSM10]. Besides friends' locations, users' profile attributes, such as hometown, school and workplace, may also indicate their current city to some extent [CHC13]. In order to achieve high prediction accuracy in Facebook, we devise a novel current city prediction approach by extracting location indications from integrated self-exposed information including profile attributes and friends list.

Second, based on the proposed prediction approach, we construct a current city exposure estimator to estimate the exposure probability that a user's invisible current city may be correctly inferred via his self-exposed information. The exposure estimator can also provide a user with some countermeasures to keep his hidden current city undercover. To the best of our knowledge, this is the first work that attempts to estimate a user's exposure probability of an invisible attribute by his self-exposed information.

2.2.1 Problem Statement

In this section, we formulate the current city prediction problem. Facebook, as a social network containing location information, can be viewed as an undirected graph G = (U,E,L), where U is a set of users; E is a set of edges e⟨u,v⟩ representing the friend relationship between users u and v, where u and v ∈ U; and L is a candidate locations list composed of all the user-generated locations.

Typically, a user u in Facebook might contribute various information, e.g., his basic profile information, friends, comments, photos. The core information of the user u in this research is his current city, denoted as l(u). According to the accessibility of users' current city, the users are classified into two sets: current city available users (LA-users) and current city unavailable users (LN-users). We, respectively, use U^{LA} and U^{LN} to denote the sets of LA-users and LN-users. Thus, we have $U=U^{LA} \cup U^{LN}$.

To predict users' current city, we tend to exploit the users' location sensitive attributes and friends list. Assume that there exist m types of location sensitive attributes, denoted as A = {a_1, a_2, …., a_m}. Specifically, we denote a user u's location sensitive attributes as A (u) = {a_1(u), a_2(u), …., a_m(u)}. The users may also have a friends list, denoted as F(u), where F(u) = {f ∈ U ∧ e⟨u,f⟩ ∈ E}. Therefore, we use a tuple to represent a user as u: ⟨l(u), A(u), F (u)⟩.

Additionally, we attempt to use a tuple to denote a location. In fact, each location is associated with a unified ID (i.e., l_{id}). Then, with this *ID*, we can obtain its latitude and longitude coordinate through Facebook Graph API Explorer. Herefore, a location can be written as a tuple l : <l_{id}, lat, lon> and the candidate locations list can be denoted as a set of location tuples: L = {tuple l : <l_{id}, lat, lon>$\}_N$, where lat and lon respectively stand for the latitude and longitude of a location, and N is the number of candidate locations in the list. Thus, the current *city prediction problem* can be formally stated as: Given,

- A graph G = ($U^{LA} \cup U^{LN}$, E , L)
- the public location l(u) for LA-users u ∈ U^{LA}
- the location sensitive attributes A(u) and the friends list F(u) for all the users u∈($U^{LA} \cup U^{LN}$)

we predict current city $\hat{l}(u)$ for each LN-user u ∈ U^{LN}, so as to make $\hat{l}(u)$ close to the user's real current city.

We want to note that the current city of a user's friends can be either available (f ∈ U^{LA}) or unavailable (f ∈ U^{LN}). Thus, we introduce two notations to represent the two groups of friends: current city available friends (LA-friends) and current city unavailable friends (LN-friends). Let denote a user's LA-friends as $F^{LA}(u)$ and LN-friends as $F^{LN}(u)$, where F(u) = $F^{LA}(u) \cup F^{LN}(u)$.

2.2.2 Overview of Current City Predictor

The goal of current city prediction is to correctly infer a coordinate point with latitude and longitude for a LN-user, given the candidate locations list L and the user's self-exposed information including his location sensitive attributes and friends list. To achieve this goal, the basic idea is training a unified location indication model by extracting and integrating location indications from the given self-exposed information. This trained model is expected to estimate the probability of the given LN-user being at each location in the candidate locations list. Then a prediction approach is proposed to properly select a location from the candidate locations as the predicted current city.

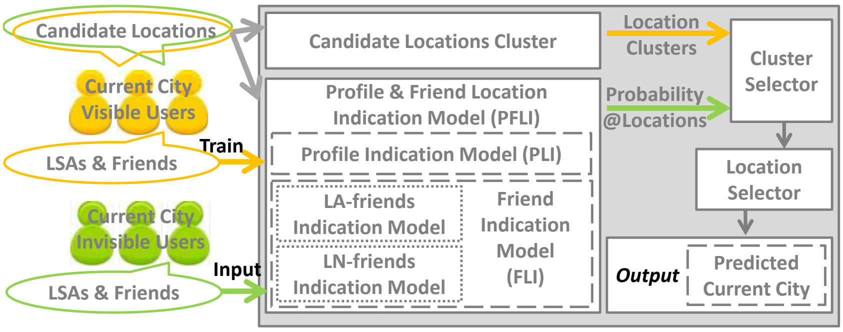

Candidate Locations
Current City Visible Users
Train
LSAs & Friends
Current City Invisible Users
Input
LSAs & Friends
Candidate Locations Cluster
Profile & Friend Location Indication Model (PFLI)
Profile Indication Model (PLI)
LA-friends Indication Model
LN-friends Indication Model
Friend Indication Model (FLI)
Location Clusters
Probability @Locations
Cluster Selector
Location Selector
Output
Predicted Current City

Figure 13. Framework of current city prediction

Figure 13 illustrates the framework of our proposed current city prediction approach. We separately consider the location indications from location sensitive attributes and friends, and consequently train two sub-models: profile location indication (PLI) model and friend location indication (FLI) model. Both PLI model and FLI model calculate a probability vector in which the element stands for the probability of a user being at a certain candidate location. Note that, FLI model leverages the location indications from both LA-friends and LN-friends. By summing up the probability vectors generated by PLI and FLI model with proper parameters, a unified profile and friends location indication (PFLI) model is derived. This PFLI model gives an integrated probability vector and tells the probabilities that a user may currently lives in the candidate locations.

With the candidate locations and the corresponding probabilities, we use a two-step location selection strategy — including cluster selection and location selection — to predict a user's current city. First, we aggregate the nearby locations into a location cluster. Then, we calculate the probability of a user being in a cluster by summing up the probabilities of all the candidate locations belonging to this cluster. Subsequently the cluster with the highest probability is picked out as a candidate cluster. Finally, we try to select the 'best' location from the candidate cluster as the predicted current city.

In the next two sections, we are going to introduce how we set up the PFLI model and devise the current city prediction approach in detail.

2.2.3 Profile and Friend-location Indication Model

In this section, we go into details about the design of the probabilistic models which can point out the probabilities of users being at all candidate locations. We first introduce the profile location indication (PLI) model which estimates the probability merely relying on a user's location sensitive attributes. Then, we describe the friend location indication (FLI) model depending on a user's friends. Eventually, we integrate these two models together and obtain the integrated profile and friend location indication (PFLI) model.

2.2.3.1 Profile Location Indication Model

Two problems are concerned to construct PLI model. First, multiple location sensitive attributes may be captured from a user's profile. This requires us to integrate the location indications extracted from different location sensitive attributes; Second, one value of a location sensitive attribute may indicate several locations. Therefore, for each attribute value, we need to consider its all possible location indications with the corresponding probabilities.

In order to capture the multiple possible location indications from one attribute value, we first define a *location-attribute indication matrix* for each (k-th) location sensitive attribute $a_k \in A$, denoted as R_k. The rows of this matrix represent the candidate locations (i.e., $l \in L_N$). We use l_i while the columns stand for the possible values of a_k. We use l_i to represent the i-th candiate location a_{kj} to denote the j-th possible value of a_k. Then, a cell in σ_k^{ij} in the matrix calculates the indication probablity of a_{kj} to l_i — the probability a user, whose k-th location sensitive attribute (a_k) equals a_{kj}, currently lives in the city $l_i \in$ L. Specifically, it equals the number of users who live in l_i and have a value of a_{kj} divided by the total number of users who have a value a_{kj}. For instance, considering workplace, if 10 out of 100 employees from TELECOM SUDPARIS state that they live in EVRY in the whole dataset, then the indication probability of TELECOM SUDPARIS to EVRY is 0.1. Moreover, if we take one column (e.g., j-th) out of the R_k, it represents the multiple location indications of a_{kj}.

Assume that a_k refers to M possible values except for 'null'; N is the total number of the candidate locations. The location attribute indication matrix can be written as:

$R_k = \{\sigma^{ij}\}_{N\times M} = \{p(l(u) = l_i \,|\, a_k(u) = a_{kj}\,)\}_{N\times M}$

Based on the above location-attribute indication matrix (R), we model the probability of a user's current city at $l_i \in$L by combining all the user's available location sensitive attributes in his profile:

$$
\begin{aligned}
p_{Prof}(u,l_i) &= \underset{a_k\in\mathcal{A},a_k(u)\neq null}{} \alpha_k p(l(u)=l_i|a_k(u)=a_{k_j}) \\
&= \underset{a_k\in\mathcal{A},a_k(u)\neq null}{} \alpha_k \sigma_k(a_k(u),l_i)
\end{aligned}
\qquad (1)
$$

where $\sigma_k(a_k(u), l_i)$ can be easily obtained by indexing the corresponding location-attribute indication matrix (R_k) according to u's value a_k (a_{kj}) and the given location (li), namely σ_k^{ij}; a_k is a parameter to adjust the significance of the different location sensitive attribute.

Not all attributes of a user can be observed by public. Therefore, in Eq. 1, we merely consider the location sensitive attributes where the user publishes a value (i.e., $a_k(u) \neq$ null). That means the indication probability of a user at any location is equal to zero if the user's attribute $a_k(u)$ is invisible. In addition, if all the location sensitive attributes are invisible for a user, we rely on the other information (e.g., his friends) to infer his current city, which we will discuss in the next section.

2.2.3.2 Friend Location Indication Model

Besides a user's location sensitive attributes, a plenty of location related information can be extracted from the user's friends. The existing work points out that the locations of around 92% of the crawled users from Twitter are also revealed in their relationships [LWC12]; We can find 87% of users' current city from their friends' locations in our crawled Facebook dataset. Hence, in this section, we exploit the location indication from users' friends to construct a FLI model.

A user's friends can be either a LA-friend (current city available) or a LN-friend (current city unavailable). The existing work focuses on a user's LA-friends and uses these LA-friends' locations to infer the user's location [BSM10][LWD12]. Since various friends may indicate different places, and sometimes these places may be far apart; then, the question is how to distinguish the various friends and assign a higher weight to the friends who may live closer to the user. Besides, another question is raised to understand whether the LN-friends can help to predict the user's location.

To answer the above two questions and construct an accurate FLI model, let us look at two examples shown in the Figure 14, where we have three LA-users u3, u4, u5 and two LN-users u1, u2. First, concerning a LN-user and his various LA-friends, we expect to find a clue from the examples to distinguish the LA-friends' significance so as to improve the current city prediction for the LN-user. Focusing on LN-user u2 and his LA-friends u3, u4 and u5, we notice that u2 and u3, u4 work in the same institute, while u5 works in another company which is very far away (different cities) from u2's workplace. In this case, it is natural to infer that u2 is more likely to be living in the same city with u3 and u4 than with u5; then u3 and u4 should be assigned with higher weights than u5 because of the location similarity indicated by their workplace. This example inspires us to measure a friend's weight based on the location similarity indicated by the user and the friend's location sensitive attributes.

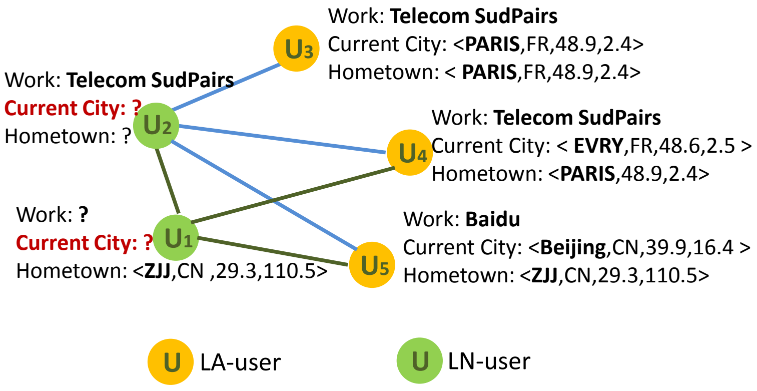

Work: **Telecom SudPairs**
Current City: <**PARIS**,FR,48.9,2.4>
Hometown: < **PARIS**,FR,48.9,2.4>
U3
Work: **Telecom SudPairs**
Current City: ?
Hometown: ?
U2
Work: **Telecom SudPairs**
Current City: < **EVRY**,FR,48.6,2.5 >
Hometown: <**PARIS**,48.9,2.4>
U4
Work: **?**
Current City: ?
Hometown: <**ZJJ**,CN ,29.3,110.5>
U1
Work: **Baidu**
Current City: <**Beijing**,CN,39.9,16.4 >
Hometown: <**ZJJ**,CN,29.3,110.5>
U5
U LA-user
U LN-user

Figure 14. An example: visualized social relations and profile information

Second, we attempt to inspect the possible utility value of a LN-user's LN-friends for his current city prediction. In Figure 14, we observe that u2, being a LN-friend of u1, does not expose his current city; whereas, the workplace of u2, TELECOM SUDPARIS, indicates two cities — PARIS and EVRY — according to the current cities of the users u3 and u4 who are also the employees of TELECOM SUDPARIS. From this example, we notice that the LN-friends also reveal some location indications in their exposed attributes, which may help the prediction.

Based on the two-side observations from the examples, FLI model takes into account the location indications from both a user's LA-friends and LN-friends. On one hand, FLI model is constructed primarily depending on the location indications from LA-friends; on the other hand, we also consider the location indications from LN-friends as a small regulator to modulate FLI model. Accordingly, FLI model contains two components: LA-friends location indication model (LA-FLI) and LN-friends location indication model (LN-FLI).

1) LA-FLI Model: A straight-forward idea of a location prediction model equally treats all the friends and determines a user's current city according to the frequencies of the locations that appear in his LA-friends. However, LA-FLI model differentiates the weights of a user's LA-friends and estimates the probability of the user living in a certain candidate location (l_i) by the weighted frequency of l_i. In other words, we sum up the weights of a user' LA-friends that live in l_i to measure the possibility of the user currently being at l_i.

To bias LA-friends' weights, we construct an attribute-based location similarity matrix (e.g., W_k) to estimate the location similarity between two users in terms of each (k-th) location sensitive attribute ($a_k \in A$). The rows and columns in the matrix are the possible values regarding a_k. The cells in the matrix (i.e., w_k^{ij}) calculate the location similarity of two users— the probability that the two users live in the same city — when they respectively have values of a_{ki} and a_{kj}. Specifically, we compute the total number of friend pairs where one user has a value of a_{ki} and the other has a value of a_{kj}, denoted as $|\{a_k (u) = a_{ki} \wedge a_k (v) = a_{kj} \}|$. Among these friend pairs, we further count the number of friend pairs where the two users live in the same city, denoted as $|\{l(u) = l(v) \wedge a_k(u) = a_{ki} \wedge a_k(v) = a_{kj} \}|$. Then, the attribute-based location similarity matrix is defined as:

$$\begin{aligned}\mathcal{W}_k &= \{w_k^{ij}\}_{M\times M}\\ &= \{p(l(u)=l(v)|a_k(u)=a_{k_i}\wedge a_k(v)=a_{k_j})\}_{M\times M}\\ &= \{\frac{|\{l(u)=l(v)\wedge a_k(u)=a_{k_i}\wedge a_k(v)=a_{k_j}\}|}{|\{a_k(u)=a_{k_i}\wedge a_k(v)=a_{k_j}\}|}\}_{M\times M}\end{aligned}$$

where M is the number of possible values of attribute ak. Note that LA-FLI model considers the location similarity as 0 if any of the user or his LA-friend does not expose attribute a_k. For a certain attribute a_k, assume that the user u and his LA-friend v have a value of a_{ki} and a_{kj} respectively. Then, the location similarity between u and v on a_k can be easily obtained by indexing the i-th row and j-th column of W_k, denoted as $w_k(u, v) = w_{ij}$, v ∈ FLA(u). On the basis of the location similarity on a certain attribute, we combine all the location similarities on multiple location sensitive attributes with a set of trained parameters (β_k for a_k) to measure the weight of the LA-friend v. This combined attribute-based weight describes the probability that u and v live in the same city concerning all of their location sensitive attributes (e.g., work, hometown). v will be assigned to a large weight if he has a high probability to be in the same city with u. Then, taking into account LA-friends' weights, LA-FLI model calculates the probability of u living in li ∈ L by integrating all the location indications from LA-friends and it can be written as:

$$p_{LA-F}(u,l_i) = \underset{v\in\mathcal{F}^{LA}(u)}{} \underset{a_k\in\mathcal{A}}{} \beta_k w_k(u,v) p_{LA-U}(v,l_i) \quad (2)$$

where $P_{LA\text{-}U}$ (v, l_i) represents the probability of the LA-friend v living in l_i. It equals 1 if v states his current city is l_i; otherwise, the probability is 0. We denote this probability as:

$$p_{LA-U}(v,l_i) = \begin{matrix} 1 & \textit{if } l(v)=l_i \\ 0 & \textit{others}\end{matrix}$$

2) LN-FLI Model: For a LN-friend (v), relying on his exposed location sensitive attributes, we first use PLI model to predict v's location, as:

$$p_{Prof}(v,l_i) = \underset{a_k\in\mathcal{A},a_k(v)\neq null}{} \alpha_k p(l(v)=l_i|a_k(v)=a_{k_j})$$

In addition, we regard all the LN-friends as equal and sum up all the location indications from LN-friends. Consequently, for l_i ∈ L, LN-FLI model can be written as:

$$p_{LN-F}(u,l_i) = \underset{v\in F^{LN}(u)}{} p_{Prof}(v,l_i) \quad (3)$$

Eventually, primarily relying on LA-FLI model and being adjusted by LN-FLI model with a very small regulator λ, FLI model estimates the probability that u currently lives at l_i by:

$$p_F(u,l_i) = p_{LA-F}(u,l_i) + \lambda p_{LN-F}(u,l_i) \quad (4)$$

2.2.3.3 *Integrated Profile and Friend Indication Model*

So far, we have introduced PLI model and FLI model, which abstract the probabilities of a user at various candidate locations, respectively, from his own location sensitive attributes and friends list. Then, we integrate them into a hybrid probabilistic location indication model, so as to capture complete location indications from two sides. In summary, PFLI model calculates the probability of u at $l_i \in L$ as:

$$p(u, l_i) = \theta_P p_{Prof}(u, l_i) + \theta_F p_F(u, l_i) \qquad (5)$$

Parameter Computation: To obtain a set of good parameters for the model, we first write the model in the following way:

$$\begin{aligned} p(u, l_i) &= \theta_P p_{Prof}(u, l_i) + \theta_F p_F(u, l_i) \\ &= \quad \underset{a_k \in \mathcal{A}}{} \quad \theta_P \alpha_k \sigma_k(a_k(u), l_i) \\ &+ \quad \underset{a_k \in \mathcal{A}}{} \quad \theta_F \beta_k \quad \underset{v \in F^{LA}(u)}{} \quad w_k(u, v) p_{LA-F}(v, l_i) \\ &+ \quad \underset{a_k \in \mathcal{A}}{} \quad \lambda \theta_F \quad \underset{v \in F^{LN}(u)}{} \quad \alpha_k \sigma_k(a_k(v), l_i) \\ &= \quad \underset{a_k \in \mathcal{A}}{} \quad \{[\mu_k \sigma_k(a_k(u), l_i) + \nu_k \delta_k(u, l_i)] \\ &+ [\lambda_\alpha \eta_k(u, l_i)]\} \end{aligned} \qquad (6)$$

where,

- $\mu_k = \theta_P \alpha_k; \; \nu_k = \theta_F \beta_k; \; \lambda_\alpha = \lambda \theta_F$
- $\delta_k(u, l_i) = \quad {}_{v \in F^{LA}(u)} \, w_k(u, v) p_{LA-F}(v, l_i)$
- $\eta_k(u, l_i) = \quad {}_{v \in F^{LN}(u)} \, \sigma_k(a_k(v), l_i)$

As λ is a regulator of very small value, we take the location indications extracted from a user's location sensitive attributes and his LA-friends as primary indications, while the part captured from the LN-friends as a micro-regulating indication. Thus, we compute the parameters by two separate steps. We first optimize the parameters μ_k and ν_k together for the main indication. We also try to find a set of local optimal parameters for PLI model so as to have a good regulating value from the LN-friends.

We use LA-users as a training dataset to obtain a group of good parameters. We consider all the locations that indicated either by the user's location sensitive attributes or his friends and classify these indicated locations into two groups by their distances to u's actual location. If the distance is larger than a certain value we label the corresponding indicated location as a far location (i.e., label = 0); otherwise, we regard it as a close location (i.e., label = 1). Based on each indicated location of the user u, we generate an independent item ⟨label, σ_k(u,li), δ_k(u,li)⟩. We apply these items to construct a logistic regression model in the format of $f(y|x; \theta) = h_\theta(x)^y(1 - h_\theta(x))^{1-y}$, where y is the label of the indicated location, x is either the value of σ_k(u,li) or of δ_k(u,li), and $h_\theta(x)$ is the hypothesis function. Then we can apply gradient descent method to maximize $f(y|x;\theta)$ and compute the parameters. Similarly, we can obtain a set of parameters for PLI model which works for the LN-friends.

Model Adjustment: Concerning the incompletion of users' self-exposure information, the proposed PFLI model needs to fully satisfy all types of users. For users who hide their friends lists from public, PFLI model only relies on the users' location sensitive attributes and degenerates into PLI model. Similarly, PFLI model is simplified into FLI model when all of the users' location sensitive attributes are unobserved.

Besides, for a completely latent user (i.e., $u^\emptyset$) who exposes neither location sensitive attributes nor friends, we take two steps: (S1) we try to capture the friends of $u^\emptyset$ in reverse from other users who expose their friends. The implicit principle is derived from the indirect friend relationship — v is also a friend of $u^{\emptyset}$ if $u^{\emptyset}$ is a friend of v. That is to say, user v could be referred to as a friend of $u^{\emptyset}$, if $u^{\emptyset}$ is in v' friends list. Thus, we can apply FLI model to uØ once we can determine some friends of $u^{\emptyset}$. (S2) For the $u^{\emptyset}$ whose friends cannot be detected by S1, we use a global frequency probability model:

$$\bar{P}_{u^{\emptyset}}(l_i) = \frac{\bar{freq}(\bar{l}_i)}{\sum_{l_i \in \mathcal{L}} freq(l_i)}$$

2.2.4 Current-City Prediction Approach

Based on the results of PFLI model — the corresponding probabilities of a user currently living in all the candidate locations, we devise a current city prediction approach in this section. Recall the example we illustrated in Challenge 3 of Sec. I. Assuming the PFLI model suggests that a user u has a probability of, respectively, 40% in BEIJING, 35% in PAIRS and 25% in EVRY which is very close to PARIS. In this case, u might live in the area around PAIRS and EVRY with a larger probability than BEIJING since the aggregated probability of u in the area around PAIRS and EVRY are higher than BEIJING. Therefore, rather than directly deal with the problem on a single-city [BSM10][LWC12][CCL12][AKT12], we aggregate the candidate locations which are very close to each other into a location cluster. We attempt to predict a user's current city by two steps: cluster selection and location selection. Refer to Figure 13, this prediction approach has been implemented with several main functions, including Candidate Locations Cluster, Cluster Selector and Location Selector. We are going to explain the main functions and illustrate how the prediction approach performs.

2.2.4.1 Candidate Location Cluster

We draw on hierarchical clustering method [HTF01] to generate location clusters. The hierarchical clustering method arranges all the candidate locations in a hierarchy with a treelike structure based on the distance between two locations, and successively merges the closest locations into clusters. Specifically, we first treat all the candidate locations as an independent location cluster and calculate the distance between any two candidate locations (Step 1). We find the closest pair of the location clusters and merge them into a new location cluster (Step 2). Then, we compute the average distance between the new cluster and each of the old ones (Step 3). We repeat the Step 2 and Step 3 until all the candidate locations are organized into one cluster tree. Eventually, we choose an ideal distance threshold (i.e., the average distance between any of two locations in the neighboring location clusters) to cut the cluster tree into clusters (Step 4).

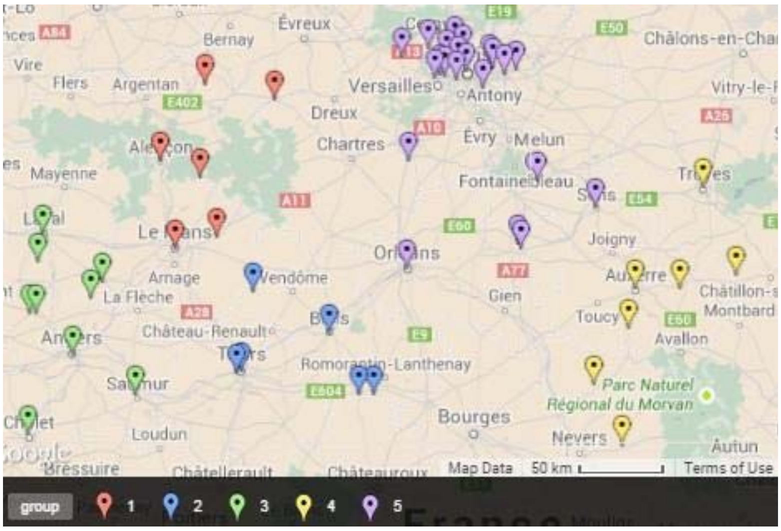


Figure 15. Example of candidate locations cluster

Figure 15 illustrates an example of clusters on the user-generated candidate locations that locate in the area with latitude in 47ºN ~ 49ºN and longitude in 1ºW ~ 6ºE. There exist 154 candidate locations in this area mentioned by users in our Facebook dataset. With the hierarchical clustering method, we divide them into 7 location clusters which are marked in different color. We note several properties of our candidate locations clusters. First, all the regions are formed by clustering the user-generated locations according to their distances, instead of dividing areas with equal-sized grid cells [LNK05] [HTF01]. In our clustering, the areas that the users do not mention are out of consideration. Second, the densities inside the clusters are different. However, the average distances between all the candidate locations in any two neighboring clusters are equal (100km in Figure 15). Third, the complexity of the Step 2 and Step 3 is O(k^3) and the complexity of the Step 4 is O(2^k). Although the computation of location clustering is expensive, it can be preprocessed and only needs to be run once.

2.2.4.2 Cluster Selector

Cluster selector selects the best cluster for a user where the user may reside with the highest probability. We leverage the proposed PFLI model to obtain the user's probability living at each candidate location. The probability of a user locating in a cluster is equal to the aggregated probability of all the candidate locations inside the cluster. Therefore, for each cluster, we sum up the probabilities of its candidate locations and select the cluster with the highest probability.

2.2.4.3 Location Selector

Eventually, we tend to select a best point from the selected cluster for the user. Three alternatives are considered here. First, we select the point with the highest probability (i.e., point of highest probability) inside the selected cluster as the best point. Second, we consider the geographic centroid of the selected cluster as the user's best point. The geographic centroid is the average coordinate for all the points in a cluster while the probability of each point is considered as its weight. Third, we calculate the center of minimum distance which minimizes the overall distance form itself to all the rest of locations in a cluster. We will further discuss and compare the three methods in the experiment.

2.2.4.4 Implementation of Prediction Approach

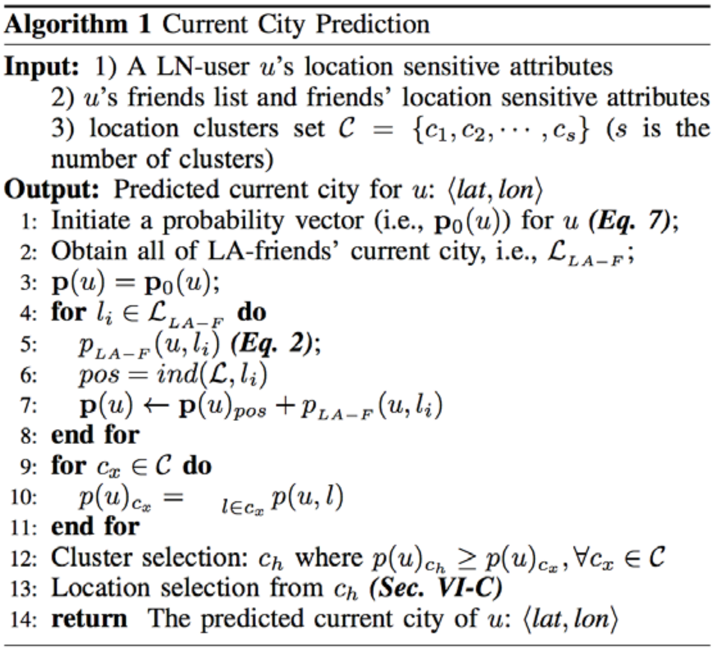
Practically, each user only associates with a very limited number of locations compared to the total number of the candidate locations. Hence, according to the PFLI model, we do not calculate the probability for each candidate location and simplify the computation by three steps.

First, we initiate a probability vector (i.e., p_0(u)) of candidate locations for a user, which merely includes the location indications from the user's location sensitive attributes and his LN-friends. Recall that the j-th column in the location-attribute indication matrix R_k stands for the probabilities that a user lives in the corresponding locations if the user presents a value of a_{kj} in terms of the attribute a_k. We rewrite the indication matrix as R_k = [R_{k1} , R_{k2} , ..., R_{kM}], where M is the number of possible values of a_k. Thus, we obtain the following initial probability vector:

$$\begin{aligned}\mathbf{p}_0(u) &= \underset{a_k \in \mathcal{A}}{} [\mu_k \mathbf{p}(a_k(u) = a_{k_j}) + \lambda_\alpha \underset{v \in F^{LN}}{} \mathbf{p}(a_k(v) = a_{k_j})] \\ &= \underset{a_k \in \mathcal{A}}{} (\mu_k R_{\cdot k_j} + \lambda_\alpha \underset{v \in F^{LN}}{} \alpha_k R_{\cdot k_j})\end{aligned} \tag{7}$$

Second, we look at the location indications from the user's LA-friends. In fact, such indications correspond to two factors: LA-friends' current city and their weight. The current city of the user's LA-friends are aggregated into a set, denoted as L_{LA-F} . According to the LA-FLI model, we can compute the probabilities that the user lives in li∈L_{LA-F} . As the number of the locations in L_{LA-F} is much smaller than the total number of candidate locations in L, we can dramatically improve the computation rate.

Eventually, obtaining the probabilities of u in all possible locations, we can easily compute the aggregated probability of u in each cluster and select the cluster with the highest probability. Furthermore, we predict a current city for u from the selected cluster by exploiting the location selector.



```
Algorithm 1 Current City Prediction
Input: 1) A LN-user $u$'s location sensitive attributes
    2) $u$'s friends list and friends' location sensitive attributes
    3) location clusters set $\mathcal{C} = \{c_1, c_2, \cdots, c_s\}$ ($s$ is the number of clusters)
Output: Predicted current city for $u$: $\langle lat, lon \rangle$
Initiate a probability vector (i.e., $\mathbf{p}_0(u)$) for $u$ (Eq. 7);
Obtain all of LA-friends' current city, i.e., $\mathcal{L}_{LA-F}$;
$\mathbf{p}(u) = \mathbf{p}_0(u)$;
for $l_i \in \mathcal{L}_{LA-F}$ do
    $p_{LA-F}(u, l_i)$ (Eq. 2);
    $pos = ind(\mathcal{L}, l_i)$
    $\mathbf{p}(u) \leftarrow \mathbf{p}(u)_{pos} + p_{LA-F}(u, l_i)$
end for
for $c_x \in \mathcal{C}$ do
    $p(u)_{c_x} = \quad _{l \in c_x} p(u, l)$
end for
Cluster selection: $c_h$ where $p(u)_{c_h} \geq p(u)_{c_x}, \forall c_x \in \mathcal{C}$
Location selection from $c_h$ (Sec. VI-C)
return The predicted current city of $u$: $\langle lat, lon \rangle$
```

Figure 16. Algorithm

We summarize the current city prediction approach in Figure 16. In this algorithm, we assume that we have acquired the trained PFLI model with parameters (μ_k, ν_k, λ_k), the candidate locations (L), the location attribute indication matrices (R_k), the attribute-based location similarity matrix (W_k) and the location clusters set (C). Then we try to predict the user u's current city according to u's self-exposed information. Finally, we obtain a location with its latitude and longitude.

2.2.5 Empirical Evaluation

In this section, we evaluate our proposed approach on a dataset crawled from Facebook. We first introduce this dataset, the compared approaches and the measurement. Then, we report the experiment results.

2.2.5.1 Experimental Setup

We crawled Facebook by a Bread First Search (BFS) [GKB11] strategy from March to June in 2012 and collected social information from 371,913 users including their profiles and friends lists. Among all these users, 153,909 users publicly report their current city (LA-users) and the rest 225,314 users do not reveal their current city (LN-users). In our evaluation, we use the 153,909 LA-users as the train and test set for the prediction. All the LN-users' information is also involved in the experiments, as the proposed approach considers the integrated location indications not only from a user' location sensitive attributes and LA-friends but also from LN-friends. We extract a user's latest work or education experience as a location sensitive attribute, named 'Work and Education'; we also exploit a user’s ‘Hometown’ as another location sensitive attribute. In our dataset, 122,899 LA-users show their ‘Hometown’ and 54,097 LA-users expose ‘Work and Education’ to the public. In addition, 115,807 of the LA-users publish their friends list.

Based on our current city prediction model and the two-step location selection strategy (cluster selection; then location selection), we propose three cluster based prediction approaches with different location selectors. We tend to compare the performance of these approaches and determine a good location selector to obtain a prediction approach with high prediction accuracy. We also propose a non-cluster prediction approach based on our current city prediction model to evaluate the effectiveness of location cluster. Specifically, these model based approaches can be denoted as:

- $PFLI_{prob}$ is a cluster based approach which selects the point of highest probability from the selected cluster as the predicted location.
- $PFLI_{cent}$ is a cluster based approach which selects the geographic centroid from the selected cluster as the best prediction.
- $PFLI_{dist}$ is a cluster based approach which selects the center of minimum distance from the selected cluster.
- $PFLI_{noclst}$ is a non-cluster approach which selects the point of highest probability from all candidate locations as the predicted location.

Besides, we compare the model based approaches with several state-of-the-art prediction approaches:

- $Base_{ann}$ maps any location sensitive attribute value to a certain location and apply artificial neural network to train a current city prediction model [CHC13], which is our previous work.
- $Base_{freq}$ infers a user’s location according to the location frequency extracted from his friends’ location-specific tweets [CKM11][CCL12]. We borrow the idea of counting the frequency of locations that emerge in a user’s friends and predict his current city by the most frequent location.
- $Base_{knn}$ also relies on the frequency idea for Twitter; however, it merely counts on a user’s k closest friends who have the most common friends with him to compute the most frequent location [AK10][AKT12].
- $Base_{dist}$ predicts a user's location based on the observation that the likelihood of friendship between two persons is decreasing with the distance [BSM10]

Among the above approaches, $Base_{dist}$ and $Base_{ann}$ are originally devised for Facebook; while $Base_{freq}$ and $Base_{knn}$ for Twitter. We leverage the main ideas from $Base_{freq}$ and $Base_{knn}$, and modify them to fit our dataset. By comparing our approach to $Base_{dist}$, $Base_{freq}$ and $Base_{knn}$ which mainly depend on friendships, we attempt to reveal the advantage of our approach: integrating location sensitive attributes. Using $Base_{ann}$, we verify the newly introduced one-attribute/multiple-locations mapping method.

We exploit the same measurements as used in the existing work [CKM11][CCL12][LWD12]: Average Error Distance (AED) and Accuracy within K km (ACC@K).

Error Distance of a user u’s predicted result (i.e., ErrDist(u)) is defined as the distance in kilometers between the user’s real location and his predicted location. AED averages the Error Distances of the overall evaluated users denoted as,

$$AED = \frac{\sum_{u \in U} ErrDist(u)}{|U|}$$

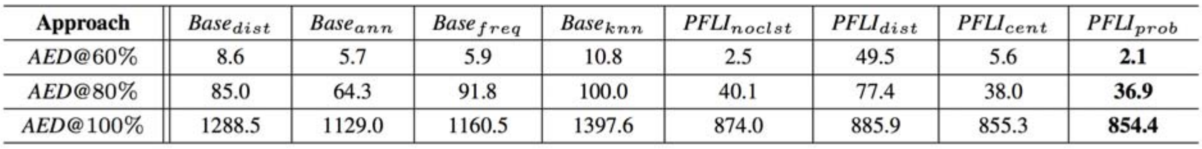
In addition, we rank the users by their Error Distance in descending order and report AED of the top 60%, 80% and 100% of the evaluated users in the ranking list, denoted as AED@60%, AED@80% and AED@100% respectively [16]. Accuracy within K km reveals the percentage of users being predicted with an Error Distance less than K km. It can be represented,

$$ACC@K = \frac{|\{u | u \in U \wedge ErrDist(u) < K\}|}{|U|}$$

ACC@K shows the predication capability of an approach at a specific required Error Distance.

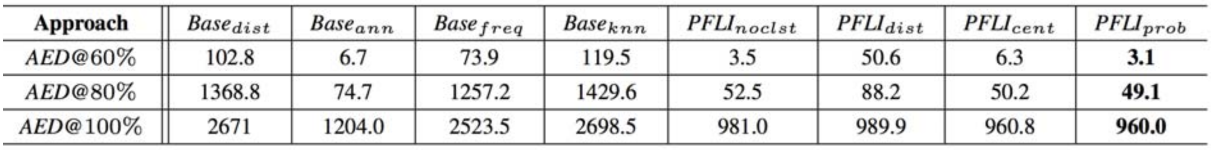
2.2.5.2 Evaluation

Many relationship-based methods (e.g., $\text{Base}_{\text{dist}}$, $\text{Base}_{\text{freq}}$ and Base_{knn}) heavily rely on users' LA-friends whose locations are exposed. In general, such methods can work well for the users who have a certain number of LA-friends. But when they are applied to the overall users (either have or have not LA-friends), the performance decreases notably. We evaluate the prediction performance respectively on two user sets: users with LA-friends and overall users. And we report the evaluation results on AED and ACC@K subsequently.



Approach	$Base_{dist}$	$Base_{ann}$	$Base_{freq}$	$Base_{knn}$	$PFLI_{noclst}$	$PFLI_{dist}$	$PFLI_{cent}$	$PFLI_{prob}$
$AED@60\%$	8.6	5.7	5.9	10.8	2.5	49.5	5.6	**2.1**
$AED@80\%$	85.0	64.3	91.8	100.0	40.1	77.4	38.0	**36.9**
$AED@100\%$	1288.5	1129.0	1160.5	1397.6	874.0	885.9	855.3	**854.4**

Table 4. Prediction results (AED) for users with LA-friends.



Approach	$Base_{dist}$	$Base_{ann}$	$Base_{freq}$	$Base_{knn}$	$PFLI_{noclst}$	$PFLI_{dist}$	$PFLI_{cent}$	$PFLI_{prob}$
$AED@60\%$	102.8	6.7	73.9	119.5	3.5	50.6	6.3	**3.1**
$AED@80\%$	1368.8	74.7	1257.2	1429.6	52.5	88.2	50.2	**49.1**
$AED@100\%$	2671	1204.0	2523.5	2698.5	981.0	989.9	960.8	**960.0**

Table 5. Prediction results (AED) for overall users

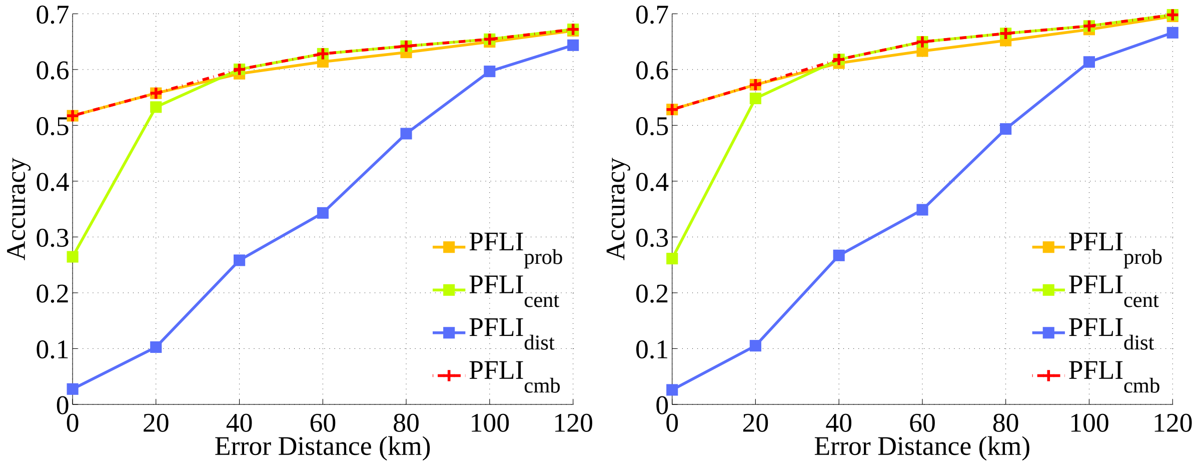
1) Evaluation on AED: Table 4 and Table 5 show the AEDs of all the compared approaches for two user sets respectively. We use bold font to highlight the shortest AED in the tables. From the results, we observe that the approaches based on our proposed PFLI model perform much better than all the other baselines. Among the model-based approaches, $\text{PFLI}_{\text{prob}}$, which selects the point with highest probability from the selected cluster, generates less AED than the other approaches with different location selectors; whereas the differences are quite small among all these model-based approaches. For instance, comparing $\text{PFLI}_{\text{prob}}$ and $\text{PFLI}_{\text{cent}}$, the differences of AED@100% are only 0.9 and 0.8 km for two user sets respectively. However, $\text{PFLI}_{\text{prob}}$ reduces the AED significantly compared to Base_{ann} — the best baseline. This observation demonstrates that our integrated probabilistic model can better describe users' location than the other compared models.

Comparing the results of AED@60%, AED@80% and AED@100%, we notice that we can predict the top 60% and the top 80% of the users' current city at relatively small AEDs by our proposed approaches; while the AEDs increase by 10-23 times when we consider all of the users (AEDs@100%). It is similar when it comes to the other approaches: AED@100% is much larger than AED@60% and AED@80%. From the perspective of the approaches' capacity, this observation demonstrates that the approaches can predict most of the users' current city with a small Error

Distance. While from the perspective of privacy, it implies that many users may be not security enough to hide their current city. We will discuss it further in the next section.

In addition, we notice that the AEDs of $Base_{dist}$, $Base_{freq}$ and $Base_{knn}$ for the overall users are almost 2 times the values for users with LA-friends. However, for the PFLI model based approach, the AEDs differ slightly for two user sets. For example, the AED of $PFLI_{prob}$ for overall users is only 74.4 km larger than the result for users with friends. Based on the evaluation comparisons on two users set, we can tell that, with the integrated location indications from users' profile and friends, our proposed prediction approaches is not constrained to users' LA-friends. Even for some users without knowing LA-friends in the overall users, our proposed approaches can still predict their location based on their profile and LN-friends. Lastly, we compare the AEDs of the approaches using our proposed PFLI model. First, we compare $PFLI_{noclst}$ and $PFLI_{prob}$. $PFLI_{noclst}$ directly selects the location of the highest probability from the probability vector generated by PFLI model; while, relying on a cluster strategy, $PFLI_{prob}$ successively takes a cluster selection and a location selection which selects the location of the highest probability inside a selected cluster. The experiment results demonstrate that the cluster based approach outperforms the non-cluster based approach. Second, we investigate the cluster based approaches with different location selection solutions. From the results, $PFLI_{dist}$ generates the largest AEDs and $PFLI_{prob}$ achieves the smallest ones. This may suggest us a good solution — selecting the point with the highest probability — to select a location inside a cluster. We will further compare these three approaches on ACC@K and determine a good location selection solution to achieve a prediction approach with high accuracy in the next section.

2) Evaluation on ACC@K: In this section, we first study ACC@K of the three proposed prediction approaches with different location selectors, attempting to understand their strengths. Based on this study, we will develop a combined approach strategy by combining the best prediction approaches under certain conditions, so as to obtain better performance than solely using any one of them.

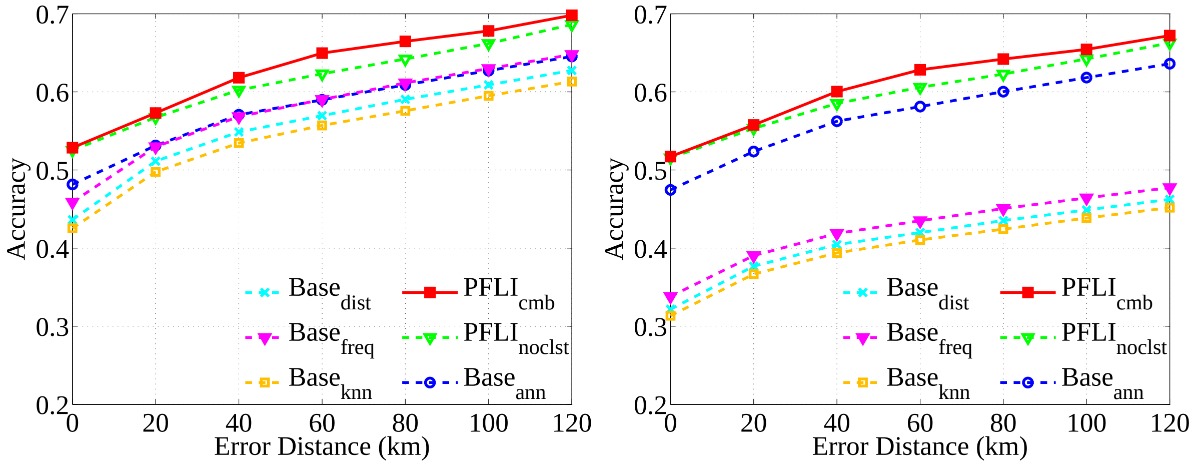


(a) Users with friends (b) Overall users

Figure 17. ACC@K comparison among different location selectors

Figure 17 compares the three proposed prediction approaches and plots ACC@Ks at different Error Distances for two user sets in two subfigures. In both subfigures, we observe that the accuracy of $PFLI_{prob}$ goes up steadily with the increase of Error Distance. Compared to $PFLI_{prob}$, $PFLI_{cent}$ may lead to very low accuracy when the required Error Distance is quite small; but it can achieve higher accuracy than $PFLI_{prob}$, when the Error Distance is larger than 40 km. It reveals the properties of these two prediction approaches: $PFLI_{cent}$ selects the geographic centroid of a cluster, which generates a short average Error Distance to all the locations in the cluster but loses chance to pick the user's exact coordinate once it is not the centroid; while $PFLI_{prob}$ might produce a large Error Distance if the location of the highest probability is not the user's real location. Besides, PFLI_{dist} is not competitive with the other two approaches.

In this case, we propose a combined-approach strategy which uses PFLI_{prob} when the required Error Distance is smaller than 40 km and otherwise applies PFLI_{cent}. We believe the combination is reasonable and practical. Because if third parties want to identify users according to their locations, they usually expect to identify users in a city or an area which allows certain Error Distance. Then, if a third party can tolerate a larger Error Distance, we can exploit PFLI_{cent}. Otherwise, we apply PFLI_{prob}. We also plot the combination line in Figure 17, named PFLI_{cmb}.



(a) Users with friends

(b) Overall users

Figure 18. ACC@K comparison between the proposed approach and other baselines

Figure 18 compares PFLI_{cmb} to various baseline methods in terms of prediction accuracy. We observe that the proposed PFLI_{cmb} outperforms all the compared baselines with the highest accuracy for both user sets. Compared to PFLI_{noclst}, PFLI_{cmb} increases around 1.5% and 1.2% of accuracy on average respectively for users with LA-friends and overall users. It proves the effectiveness of the cluster strategy with successive cluster selection and location selection. Comparing the results respectively for users with LA-friends and overall users, we observe a huge accuracy gap for Base_{freq}, Base_{dist} and Base_{knn}. These approaches severely depend on friends' locations which lead to dramatic fall of performance when they are applied for users who do not have LA-friends. However, our proposed approaches integrating location sensitive attributes and friends (including our previous work Base_{ann} [3]) can almost hold the prediction effectiveness for the overall users.

To summarize, first, we propose to combine PFLI_{prob} and PFLI_{cent} into a PFLI_{cmb} approach inspired by the experiment observations. PFLI_{cmb} can flexibly change the prediction approach according to their performance under different required Error Distances. Second, our proposed approach outperforms the other compared baselines. Especially for the overall users, our proposed approach could gain 20% higher of accuracy than Base_{dist} which is also a city prediction approach on Facebook.

2.2.6 Current-City Exposure Estimator

In this section, we pay attention to estimating the exposure probability of current city for a user who hides his current city. We formally formulate the current city exposure estimation problem as: Given,

- a graph $G=(U^{LA} \cup U^{LN}, E, L)$
- the public location l(u) for LA-users $u \in U^{LA}$
- the location sensitive attributes A(u) and the friends list F(u) for all the users $u \in (U^{LA} \cup \ U^{LN})$

- a required Error Distance K km

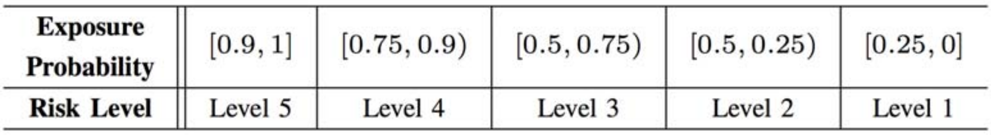
we forecast the current city exposure probability within K km and report exposure risk level for each LN-user u ∈ U^{LN} .

To solve this problem, we run the proposed prediction approach on an aggregation of users and conduct analysis on the aggregated prediction results. Furthermore, we apply a regression method to construct the exposure model according to the analysis observations. Relying on this model, we devise a current city exposure estimator to tell a user the current city Exposure Probability within K km and Exposure Risk Level.

The Exposure Probability within K km (EP@K) represents the probability that a user' current city could be inferred correctly if the required Error Distance is K km. As it has the similar concept as the metric of ACC@K, we compute it by the same formula:

$$\frac{|\{u|u\in U\wedge ErrDist(u)<K\}|}{|U|}$$

Additionally, we set up 5 Exposure Risk Level according to value of Exposure Probability, shown in Table 6. We regard Level 5 as the most risky level which indicates an Exposure Probability larger than 0.9, while Level 1 as the safe one which represents a small Exposure Probability less than 0.25.



Exposure Probability	$[0.9, 1]$	$[0.75, 0.9)$	$[0.5, 0.75)$	$[0.5, 0.25)$	$[0.25, 0]$
Risk Level	Level 5	Level 4	Level 3	Level 2	Level 1

Table 6. Risk level vs. exposure probability

Next, we first show some observations of inspections on the prediction for an aggregated user. Then we introduce the current city exposure model and the model based estimator. Finally, we illustrate some case studies to show the use of our proposed exposure estimator. We also summarize some guidelines to reduce the exposure risk.

2.2.6.1 Current City Exposure Inspection

Assume that we have run the proposed prediction approach on an aggregation of users whose current city is visible. We then obtain a collection of prediction results which includes users' self-exposed information, predicted current city and actual current city. We also develop some measurements to describe the characteristics of users' self-exposed information. Based on these prediction results, we can learn the correlation between the current city exposure probability and the measurable characteristics of users' self-exposed information. First, we classify users into diverse categories with respect to the combinations of visible/invisible properties of their location sensitive attributes and friends lists. Table 7 lists the obtained seven User Categories. User Category measures the types and amount of users' self-exposed information.

User's Visible Attributes	Abbreviation
'Hometown'	'HT'
'Work and Education'	'WE'
'Friends'	'F'
'Hometown' and 'Work and Education'	'HT+WE'
'Hometown' and 'Friends'	'HT+F'
'Work and Education' and 'Friends'	'WE+F'
'Hometown', 'Work and Education' and 'Friends'	'HT+WE+F'

Table 7. Users categories by visible attributes combination

Figure 19 inspects the Exposure Probabilities for various User Categories. From this figure, we observe that different types of self-exposed information may divulge users' current city to different extent. For instance, users in 'WE' category are normally more dangerous to disclose their current city than users in 'HT' or 'F' category. We also find that the users who publish their 'WE' (in category 'WE', 'HT+WE', 'WE+F' and 'HT+WE+F') exhibit a high Exposure Probability. This means that 'WE' is a very risky attribute to leak users' current city. The results also reveal that 'HT' is more sensitive to disclose current city than 'F', although 'F' is generally regarded as a significant location indication.

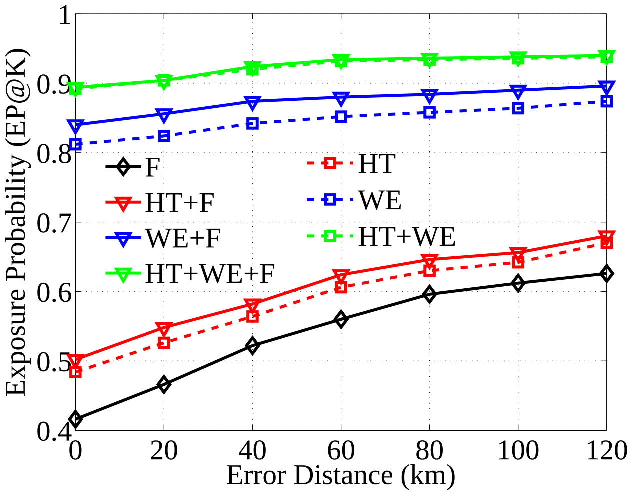


Figure 19. Current city exposure probability by user category

Besides, generally speaking, Figure 19 displays that a user's current city could be predicted with a larger probability if the user exposes more information. For example, users who expose 'HT+F' exhibit a higher exposure probability than users only revealing either 'HT' or 'F'. Note that, for a user who exposes 'WE+HT', his current city exposure probability can be up to 90% which approaches to the exposure probability of users who expose 'HT+WE+F'. In other words, merely exposing 'WE+HT' but not 'F' can almost lead to the leakage of current city.

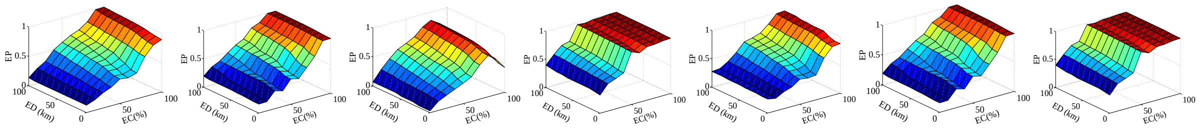
According to the results displayed in Figure 19, we conclude that User Category, distinguishing users by the types and amount of their self-exposed information, relates to Exposure Probability.

Apart from User Category, we define a new metric named Exposure Coefficient. It estimates the ratio of the probabilities of candidate locations in the selected cluster c_h to the overall probabilities of all the candidate locations (equal 1), calculated as follows:

$$EC(u) = \frac{{}_{l\in c_h} p(u,l)}{{}_{l\in\mathcal{L}} p(u,l)} = \quad {}_{l\in c_h} \; p(u,l)$$

Exposure Coefficient represents the centrality of the users' location indications. For example, Exposure Coefficient with a value of 100% means that all of a user's location indications point to an exclusive location cluster. We further look into the change of exposure probability according to Exposure Coefficient for each User Category.

Figure 20 reveals how Exposure Probability varies with diverse Exposure Coefficient and Error Distances in different User Categories. In this figure, each subfigure represents one User Category; the X, Y and Z axes in each subfigure are Exposure Coefficient (EC), Error Distances (ED) and Exposure Probability (EP) respectively. We observe that the Exposure Probability normally grows up when the Exposure Coefficient gets larger. When the Exposure Coefficient equals 100%, the Exposure Probability surpasses 90% within a required Error Distance of 20 km almost for all User Categories.



(a) HT (b) WE (c) F (d) HT+WE (e) HT+F (f) WE+F (g) HT+WE+F

Figure 20. Exposure probability by exposure coefficient in different user categories

This observation indicates that the current city is more dangerous to be predicted when a user's location indications are more likely to point to one city or to multiple cities that are in the same cluster. In other words, a user's current city is easy to disclose if the centrality of the user's self-exposed information is high.

Note that, there exists an exception for the users only exposing their 'F': the decline of Exposure Probability when the Exposure Coefficient is larger than 0.9. One reasonable explanation is that only the users with an extremely small number of friends (e.g., only 1 friend) can have an Exposure Coefficient higher than 0.9, which might reduce the risk of current city exposure due to the limited information.

2.2.6.2 Estimating Current City Exposure Risk

In the previous section, we observe that the current city Exposure Probability for a user is influenced by three factors: Error Distance, User Category and Exposure Coefficient. According to the observation which are shown in Figure 20, we try to use a polynomial multiple regression method to model the relation among the current city Exposure Probability, Exposure Coefficient and Error Distance for each User Category. We can denote the model as: $y = fun_{x1} (x_2, x_3)$, where x_1, x_2 and x_3 represent a user's User Category, Exposure Coefficient and Error Distance respectively; and $fun_{x1} (x_2 , x_3)$ represents a polynomial function of Exposure Coefficient x_2 and Error Distance x_3 given the User Category x_1. y is the computed Exposure Probability.

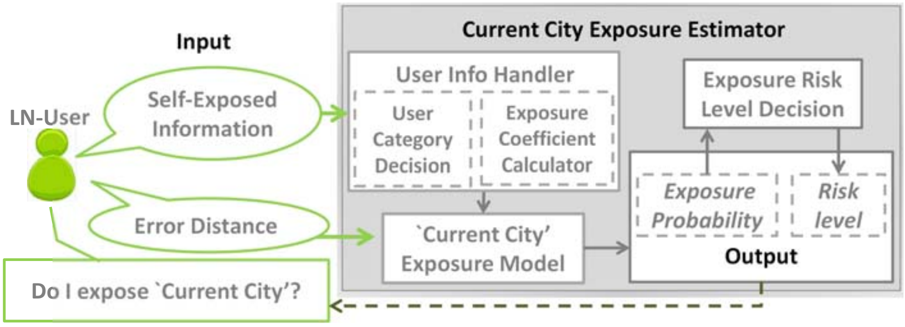

Input
LN-User
Self-Exposed Information
Error Distance
Do I expose `Current City'?
Current City Exposure Estimator
User Info Handler
User Category Decision
Exposure Coefficient Calculator
`Current City' Exposure Model
Exposure Risk Level Decision
Exposure Probability
Risk level
Output

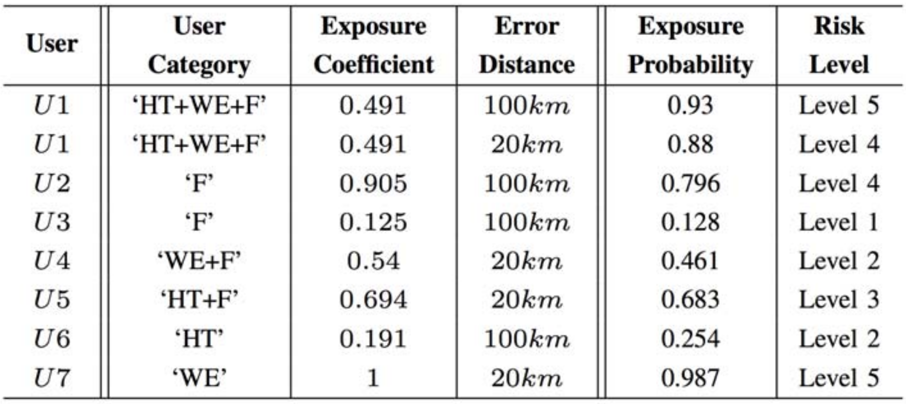
Figure 21. Framework of current city exposure estimator

By exploiting the proposed current city exposure model, we construct an exposure estimator to forecast the exposure risk of a user's private current city. Figure 21 illustrates the framework of current city exposure estimator. The exposure estimator contains three main function modules: user information handler, current city exposure model and exposure risk level decision. The inputs of the exposure estimator include a user's self-exposed information and a pre-established Error Distance. Given the user's self-exposure information, user information handler determines User category and computes Exposure Coefficient. Based on the pre-established Error Distance, the obtained User category and Exposure Coefficient, the exposure model calculates the current city exposure probability for the user. Exposure risk module determines a risk level according to the exposure probability. Eventually, the exposure estimator provides two risk measurements of current city: Exposure probability and Risk Level.

2.2.6.3 Case Studies: Exposure Estimator and Privacy Protection

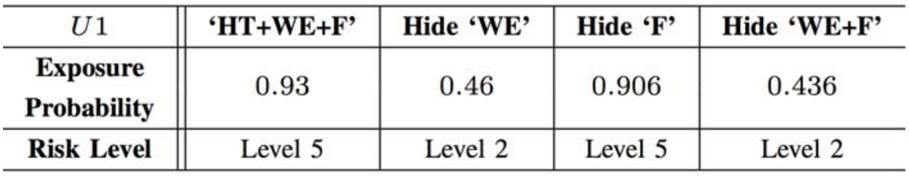
Table 8 illustrates several use cases, where we estimate the Exposure Probability and Risk Level for some LN-users. In this study, we observe that some of the LN-users are not really safe to hide their current city if they leave some other information visible. For instance, considering U7, even only publishing 'EM', his current city is almost leaked with an extremely high Exposure Probability of 0.987 within an Error Distance of 20 km. In addition, for users in the same User Category, the ones who exhibit a higher Exposure Coefficient are more likely to divulge his current city. Looking at U2 and U3 who are both in 'F' category, the current city of U2 who exhibits an extremely high Exposure Coefficients is much more dangerous than U3's current city.

In addition, the exposure estimator can give some countermeasures on privacy configuration to avoid information leakage. Assume users hide some part of their exposed information, the exposure estimator estimates and reports the corresponding Exposure Probability and Exposure Risk Level. Then users can decide a new privacy configuration accordingly. We take U1 as an example and list some possible exposure risks assuming that he adjusts his privacy configuration. The results shown in Table 9 reveal that the privacy could increase obviously if U1 hides his 'WE' or 'WE+F'. The results also point out that merely hiding 'F' could not protect U1's current city privacy.



User	User Category	Exposure Coefficient	Error Distance	Exposure Probability	Risk Level
$U1$	'HT+WE+F'	0.491	$100km$	0.93	Level 5
$U1$	'HT+WE+F'	0.491	$20km$	0.88	Level 4
$U2$	'F'	0.905	$100km$	0.796	Level 4
$U3$	'F'	0.125	$100km$	0.128	Level 1
$U4$	'WE+F'	0.54	$20km$	0.461	Level 2
$U5$	'HT+F'	0.694	$20km$	0.683	Level 3
$U6$	'HT'	0.191	$100km$	0.254	Level 2
$U7$	'WE'	1	$20km$	0.987	Level 5

Table 8. Exposure estimator cases study



$U1$	'HT+WE+F'	Hide 'WE'	Hide 'F'	Hide 'WE+F'
Exposure Probability	0.93	0.46	0.906	0.436
Risk Level	Level 5	Level 2	Level 5	Level 2

Table 9. Exposure guidelines for U1: the exposure risks if he adjusts some privacy configurations with an error distance of 100k.

Eventually, according to the studies on the current city exposure risk, we summarize the following pieces of general suggestions:

- As all the location indications may expose the hidden current city, close all of location sensitive information including 'WE', 'F' and 'HT' so as to achieve a high current city security.
- Hide the most sensitive exposed information (e.g., 'WE') if users want to publicly share some personal information (e.g., 'F'), since the most sensitive information can independently lead to a quite high Exposure Probability. For example, 'WE' alone can lead an Exposure Probability higher than 80%.
- According to the centrality principle which refers to the Exposure Coefficient, hide 'F' if most friends indicate the same place where the user lives. For instance, U2 in Table 8 is necessarily suggested hiding his 'F'.

2.2.7 Conclusion

This research starts with two open questions regarding the security of users' hidden privacy-sensitive attributes. To answer these questions, we first proposes a novel current city prediction approach to infer users' current city by leveraging users' self-exposed information including location sensitive attributes and friends list. We validate the new prediction approach on a Facebook dataset including 371,913 user and the results reveal that the users' hidden current city may be dangerous to be predicted. Then we apply the proposed prediction approach to predict users' current city and model the exposure probability to Exposure Coefficient at different Error Distances for each User Category. Based on the model, we propose a current city exposure estimator to measure the exposure probability and risk level of a user's current city according to his self-exposed information. The exposure estimator also can help users to adjust their privacy configuration to achieve their privacy intention. Note that, although this work studies the potential risk of users' privacy-sensitive attributes with a representative attribute of current city in Facebook, the proposed idea and approach could be easily extended to other attributes and utilized by other OSNs.

2.3 Characterization of Professional Publisher Activity across Twitter, Facebook and Google+

Online Social Networks (OSNs) have become one of the most popular services in the Internet attracting billions of subscribers and millions of daily active users. This tremendous success has created a golden opportunity to professional players (i.e. big industry brands, politicians, celebrities, etc.) in order to: interact with a huge amount of potential customers/voters/fans, improve their reputation and popularity, run marketing campaigns, etc. The presence and interest of professional users in OSNs as well as their concern to engage more people [WWE13] with their OSNs accounts is becoming so relevant that we can even find an award ceremony to best professionals users in social media [ASA14].

In this context there is an increasing research interest, especially in the area of management and marketing, to study what are the strategies that professional users apply in their use of OSNs [BT10, DM13, EVA10]. It seems that understanding the factors that allow professional users to engage more people with their OSN activity will have a tremendous value in the future for marketing purposes. To the best of our knowledge most of the studies available in the literature only focus on a limited number of users and extract very particular conclusions for those users that cannot be generalized. Furthermore, all previous studies are either based on manual inspection of OSNs accounts [WBL09] or interviews [WYS12] that cover very few aspects that again lead to not generalizable conclusions. Therefore, we believe that a large-scale data-driven approach based on the actual activity of a large number of professional users across major OSNs will help to shed light into the challenging problem of devising the way professional users utilize OSNs. Towards this end in this research we rely on a dataset formed by 616 very popular users with active accounts in FB, TW and G+. For each user we capture his activity (i.e., published posts) in the three systems over a long-term time window that overall generates a corpus of 2M posts.

In contrast to previous studies we do not aim at studying the strategy of individual users. Instead, our main goal is to make a global analysis to characterize the strategy of a particular sector/category (e.g., Cars Industry, Politician, Athletes, News Media, etc.) in OSNs. This analysis can be only conducted for those sectors that fulfil the following hypothesis: professional users that belong to a particular sector present a similar strategy in OSNs. Therefore, the first objective of this research is to determine whether this hypothesis is true for some sector. For this we classify the 616 users in our dataset into 62 categories according to the sector reflected by their FB account. Out of these 62 groups only 16 had enough users to perform a meaningful validation of the hypothesis. We apply the methodology proposed in [FVM13] that determines whether the behaviour of the users within a category is significantly similar and, in addition, differs from the behaviour of the users outside that category. The results reveal 8 categories whose users present a common behaviour. These categories are: Athletes, Cars, Media News, Movie, Musician-Band, News Website, Politician, and Sports Teams. After discovering 8 sectors fulfilling the baseline hypothesis, we devote our effort to derive the behavioural elements that characterize their use of OSNs.

We base our analysis in a set of meaningful behavioural elements that allow us to discriminate the strategy of each sector. These elements include: activity rate, preference among FB, TW and G+, popularity and type of content published. Using these behavioural elements we are able to describe the strategy and highlight the differential characteristics of each category. There is a last element that, to the best of our knowledge, has never been used to analyze the strategy of professional users across multiple OSNs, which is referred to as cross-posting activity. This element captures the volume of common information that a user publishes in more than one OSN. This means, when a professional user wants to post some information he can decide to publish it in a single OSN, or in multiple OSNs. Even more, when he decides to post it in multiple OSNs, there are several combinations of OSNs he could use (e.g., FB-TW or FB-G+ or TW-G+, or the three OSNs in our work). Hence, we believe that the cross-posting activity of a user is an important behavioural element that for instance reveals whether a user utilizes each OSN for different purposes or not. In this research we dedicate a full section to characterize the cross-posting phenomenon across professional users.

Finally, to conclude this research we address the very challenging question of whether the strategies implemented by each category are successful or not. To the best of our knowledge there is no standard mechanism in the literature that allows measuring the success of a strategy in OSNs. Therefore, in this research we propose a simple methodology to quantitatively measure such success. The rationale of this methodology is to estimate the number of reactions per post a category should attract based on its popularity, and compare that estimation to the actual number of reactions received by the category. We provide an estimation of the success of each category for eight types of reaction: FB Likes, FB comments, FB shares, G+ +1s, G+ reshares, TW favourite and TW retweets.

The main findings of our research can be summarized as follows:

(1) Cross-posting is a frequent practice across professional users. In addition, the cross-posting phenomenon mainly happens between FB and TW, but it is also relevant between FB and G+. However, professional users rarely publish the same information in their TW and G+ accounts.

(2) We demonstrated that for some sectors professional users present a common behaviour. The sectors we found that fulfil this statement are: Athletes, Cars, Media News, Movie, Musician-Band, News Website, Politician, and Sports Teams.

(3) Each of the categories listed above present differential elements in their use of OSNs. For instance, Athletes activity and preference is biased to TW; categories related to news are extremely active in the three OSNs; Cars is the category with major interest in G+, and Movie shows a low activity and a clear preference for FB.

(4) The categories listed above can be further clustered into three significant groups based on the similarities in their strategies: individual users (Athletes, Musician-Band, and Politician), commercial brands (Cars and Sport Teams) and news (Media News and News Website).

(5) We demonstrate that the level of engagement of a professional user is linearly correlated to his popularity, which allows us to define a model that estimates the number of reactions per post a category should obtain according to its popularity.

(6) The only categories with a successful strategy in FB are Movie (which is successful in all OSNs) and Politician, which is the only category that does not cover the engagement expectation in G+. Similarly, the only two categories that fail in attracting the expected number of reactions in TW are Media News and News Website.

2.3.1 Data Collection Methodology

In this section we explain the selection of professional OSNs, describe our crawlers to collect data from those users, and introduce the way we classify the users into categories.

The first concept we need to define is what we refer to as OSNs professional user. It corresponds to a social profile behind a private company, public body or very popular individuals that usually have presence in most of the major OSNs and pursuit different goals than regular OSN users. These professional users utilize OSNs to increase their visibility, improve their popularity, enhance their reputation, etc. Some examples of professional users include companies, celebrities, politicians, etc.

Our first challenge was to identify a numerous group of relevant professional users having active and popular accounts across FB, TW and G+. To this end, we rely on a large dataset collected for a previous work [MGF13] that includes thousands of very popular professional and regular users with an account in at least one of these OSNs. From these users we were interested in those ones that meet two requirements: (i) have an active account in FB, TW and G+; (ii) present a high popularity in at least two of the systems. We found 616 professional users that have an active account in the three systems and satisfy the popularity requirement. We validated that the selected users were actually very relevant in at least two of the three considered OSNs by means of an external source[1] that ranks professional users in each system in terms of popularity. It must be noted that in many cases the selected users appear in relatively high positions in the three rankings.

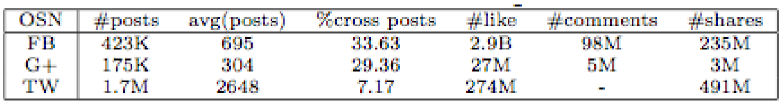
In order to define the strategies of these users we need to collect the activity of these users as well as information associated to each activity (i.e., post) like: timestamp, type of content, number of reactions, description of the post, etc. We used the crawlers developed in eCOUSIN for FB, TW and G+ for this purpose.

Following we highlight three relevant elements related to the data collection process and the implications they have for our research. (i) Our crawlers only collect public posts. However, for this particular research this is not a limiting factor since most professional publishers' posts are public. (ii) We had to convert the timestamp associated to the collected posts to a common time zone taking into account seasonal time changes. We decided to use GMT. (iii) In order to properly study the strategy of a user across FB, TW and G+ we need to use the same temporal window in the three systems. TW only allows to retrieve the last 3,200 tweets of a user that imposes a temporal limitation

[1] http://www.socialbakers.com/

that should be extrapolated to FB and G+. Then, the time window employed in each user ranges between the last collection day, 24 Aug. 2013 (which is the same for all users), and the date from which we can retrieve the oldest tweet (which varies from user to user). This guarantees an analysis of the activity for each user in the three systems during the same period.

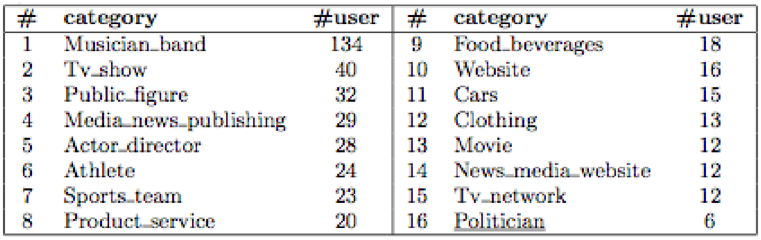
Table 10 summarizes the datasets used in this research. In total, we analyse more than 2M posts published by 616 professional publishers in FB, TW and G+.



OSN	#posts	avg(posts)	%cross posts	#like	#comments	#shares
FB	423K	695	33.63	2.9B	98M	235M
G+	175K	304	29.36	27M	5M	3M
TW	1.7M	2648	7.17	274M	-	491M

Table 10. Dataset description

Finally, in order to address the main goal of the research we need to assign the 616 users to the categories they are representing. Towards this end we have used a straightforward approach based on the category each professional user selects when they register their FB page. Therefore we assign each user to the category they have selected in FB. Overall, the 616 users are classified into 62 different categories. The goal of this research is to find whether users in some category present a common behaviour on their utilization of OSNs, describe the strategy in that category and determine its degree of success in FB, TW and G+. We can only perform that analysis for those categories in our dataset that includes enough users. Then, we have decided to study categories represented by at least 10 users in our dataset. Table 11 shows the number of users associated to the 15 categories that meet that requirement. We have made an exception for the category Politician, which is formed by only 6 users. Although we acknowledge that 6 users must not be enough to generalize the strategy of politicians, we believe it is worthy to study such an interesting category. We believe the 16 categories we are going to analyse present a quite interesting heterogeneity of sectors (e.g., popular individuals, big industrial companies, news agencies, TV or the Internet) that address different audiences.



#	category	#user	#	category	#user
1	Musician_band	134	9	Food_beverages	18
2	Tv_show	40	10	Website	16
3	Public_figure	32	11	Cars	15
4	Media_news_publishing	29	12	Clothing	13
5	Actor_director	28	13	Movie	12
6	Athlete	24	14	News_media_website	12
7	Sports_team	23	15	Tv_network	12
8	Product_service	20	16	Politician	6

Table 11. Categories in the dataset with more than 10 users.

2.3.2 Cross-Posting

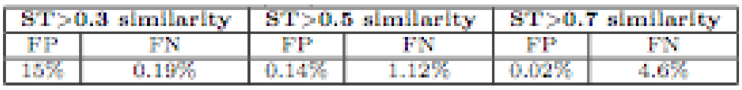
2.3.2.1 Methodology to Identify Cross-posts

In order to compare the cross-posting activity of professional users we need to have an accurate mechanism that detects when two posts are actually containing the same information. Hence, we have implemented a hierarchical classification algorithm that determines whether two posts can be considered as cross-posts in two steps. Then, given the description (i.e. the text associated with a post) of the two posts, P1 retrieved from the account of user U in OSNA and P2 published by U in his account in OSNB, our algorithm proceeds as follows:

(1) We compare P1 and P2 using NTLK Fuzzy Match [NTL13] which provides a binary decision based on the similarity of the compared texts. NTLK Fuzzy Match generates a positive answer (i.e., the same text) when both texts are very similar and only differ in some few characters. Therefore, in the context of cross-posting analysis if NTLK Fuzzy Match determines that P1 and P2 are similar, we can

safely classify them as cross-post. However, in the case in which the output is negative we cannot guarantee that P1 and P2 are not referring to the same information, thus we cannot classify them as regular posts. In summary, all the pairs of posts receiving a positive classification are labelled as cross-posts while the remaining pairs need to go through the second step of our algorithm.

(2) We compare P1 and P2 using two similarity metrics: cosine similarity [SIN01] and string similarity. These two metrics provide as output a value ranging between 0 and 1, so that the closer the output is to 1, the more similar P1 is to P2. Based on the obtained results, we classify P1 and P2 as cross-post if both metrics, cosine similarity and string similarity, are ≥ 0.5. Later in this section we validate our methodology and demonstrate why we have selected the 0.5 threshold. It must be noted that P1 is compared to P2 in case P2 was published in a period ranging between one week before and one week after P1 was published. In addition, we highlight that our algorithm is not bound to any particular alphabet so it can be applied in multiple languages.



ST>0.3 similarity		ST>0.5 similarity		ST>0.7 similarity	
FP	FN	FP	FN	FP	FN
15%	0.19%	0.14%	1.12%	0.02%	4.6%

Table 12. Validation of the cross-posts identification methodology. The table shows the false positive (FP) and false negative (FN) ratio for different similarity thresholds (ST) in percentage.

In order to ensure the accuracy of the proposed methodology 3 people manually classified 13K random posts as cross-posts or regular-posts. In order to have a meaningful validation set we ensured that half of the posts had been labelled as cross-post and half as regular-post by our classification tool. Then, given two posts published by a user in two different OSNs we classify them as a cross-post if at least 2 out of the 3 individuals performing the manual inspection indicate that both posts contain the same information. Based on the ground truth set we compute the false negative and false positive rate for our methodology using three different thresholds for the second step of the algorithm: 0.3, 0.5 and 0.7. Table 12 shows the false positive and false negative rate for our algorithm for each of the selected thresholds. The results clearly determine that 0.5 is a very good threshold since it presents a very low rate for false positives (0.14%) and false negatives (1.11%).

We applied the described methodology to the selected 616 OSN professional users and found 176K cross-posts across their OSNs accounts.

2.3.2.2 *Cross-Posting Characterization*

The first question we aim to answer is whether the cross-posting phenomenon exists in the activity of professional users, and what is its weight in FB, TW and G+. We then look at how this cross-posting occurs among the three OSNs under analysis. To this end, we quantify the fraction of cross-posting between FB-G+, FB-TW, TW-G+ and FB-TW-G+, in order to determine what set of OSNs is actually used more frequently by users to publish the same information. Finally, we also look at the preference of the users in our dataset for FB, TW and G+. We borrow the concept of preference from [25]. The authors define preference for an OSN as the bias of a user to choose more frequently that OSN as initial source of information when he aims at posting a given information in several OSNs.

2.3.2.2.1 Quantification of Cross-posting Activity

The goal is to quantify the cross-posting phenomenon for professional users in FB, TW and G+. Towards this end, we compute for each user and each OSN the portion of cross-posts with respect to all the posts each user has published. For instance, given a user U and his FB account we compute how many posts published in that account also appear in TW, G+ or both. We quantify the same parameter for the TW and G+ accounts of user U. Figure 22 shows the CDF for the portion of cross posts across the users in the three OSNs. The x axis refers to the portion of posts and the y axis to the portion of users. For instance, the point {x=0.2, y=0.4} in the line associated to FB indicates that 40% of the users have ≤20% of cross-posts in their FB accounts.

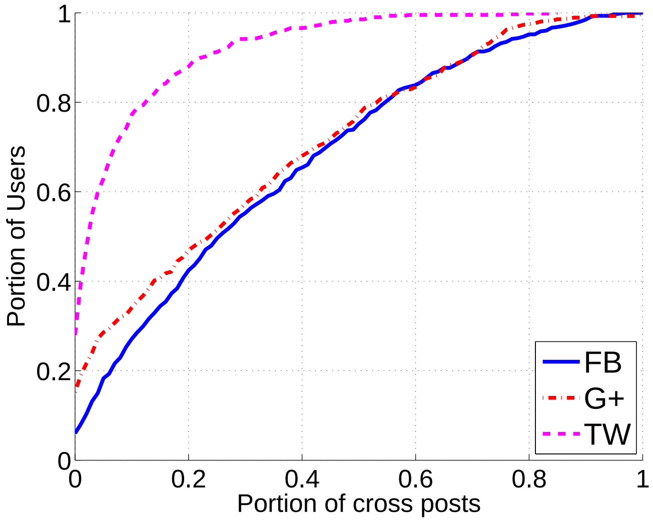


Figure 22. CDF for the portion of cross-posts per user in FB, G+ and TW.

The first immediate conclusion extracted from the graph is that most of the professional users have published some cross-posts. Only 6%, 15% and 28% of the users in FB, G+ and TW, respectively, did not present any cross-post. Hence, the first conclusion is that in general professional users find some value in cross-posting.

If we compare the results obtained for the three OSNs, we clearly observe that, in relative terms, cross-posting activity is more frequent for those posts published in FB and G+ than in TW. The results for TW show that most of the tweets are not replicated neither in FB nor in G+. The median value, which indicates the typical portion of cross-posts for a user in each OSN, shows that for a typical professional user around 1/4 of the posts that appear in FB and 1/4 of the posts that appear in G+ are also available in other OSN. However, in the case of TW, out of 100 tweets only 3 of them are replicated in other OSNs. Finally, we can find quite a large portion of professional users with intensive cross posting activity. In particular, 25%, 23% and 1.5% of the analyzed users, in FB, G+ and TW, respectively, published more cross-posts than regular-posts.

The previous analysis refers to the cross-posting activity in relative terms. However, it is important to notice that, according to the overall activity of the professional users in our dataset, the publishing rate of professional users in TW is four times higher than in FB and G+. Hence, although TW presents a much lower cross-posting activity in relative terms, it actually has a larger number of cross-posts than G+, and it is much closer to FB in the absolute number of cross-posts. In median, a professional user presents 114, 85 and 20 cross-posts in FB, TW and G+, respectively.

2.3.2.2.2 Inter-OSN Cross-posting

Once we have demonstrated that cross-posting is a common practice among professional users in FB, TW and G+, we analyze how cross-posting happens among them. Our goal is quantifying whether professional users prefer to share information in FB and TW, or rather it is more frequent finding common posts in FB and G+, or if they have more cross-posts published in TW and G+. In order to perform this analysis we proceed as follows. For a given user U we get all his cross-posts in FB (independently of whether the first appearance was in that OSN or another one) and compute which portion of them also appears in TW, which portion in G+ and which portion in both TW and G+. We repeat the same process for user U’s TW and G+ accounts. Therefore, for each user we know the cross-posting level for the following relations: FB − TW, FB − G+, TW − G+ and F B − T W − G+.

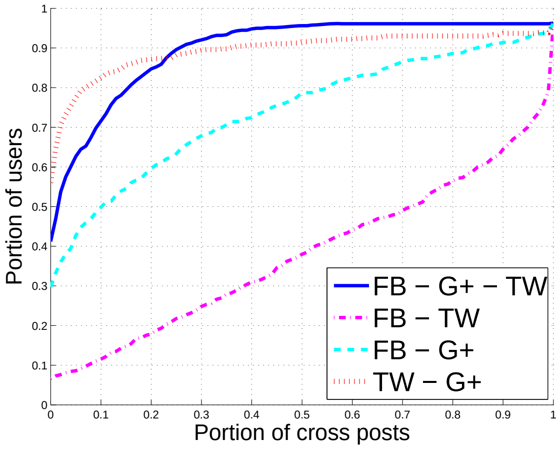
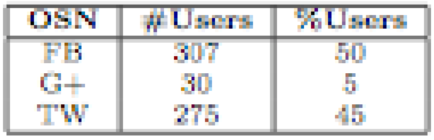


Figure 23. CDF for the portion of cross-posts and in each possible cross-posting pattern (FB-TW, FB-G+, TW-G+ or FB-TW-G+).

Figure 23 shows the CDF for the portion of cross-posts that occurs for the four referred relations across the 616 users in our dataset. Again in this figure the x axis refers to portion of posts and the y axis shows the portion of users. For instance the point x=0.4, y=0.3 in the FB-TW line indicates that 30% of the users publish ≤40% of their cross posts in FB and TW. The results reveal that professional users perform much more cross-posting between FB and TW than in any other combination of OSNs. This claim is supported by the fact that in median a professional user publishes 70% of their cross-posts on FB and TW. In addition, we find that only 8% of the users never shared a post between their FB and TW accounts, while this value grows to 30% between FB and G+, to 40% when the three OSNs are involved, and goes to 55% when we consider TW and G+. Therefore, this last result surprisingly states that is more likely that a user publishes a given information in the three OSNs than just in TW and G+.

2.3.2.2.3 Preference of Professional Publishers

We want to understand which OSN professional users prefer to publish first the information. Answering this question will roughly determine which OSN professional users value most for publishing an information that they plan to post in two or more OSNs. We define the preferred OSN of a user as the one he selected in first place for most of his cross-posts [25]. For instance, if a user has generated 20 cross-posts from which 10 were first published on FB, 6 on G+ and 4 on TW, we define FB as the preferred OSN for that user. Table 4 shows the number and portion of users in our dataset that prefer each OSN. The results reveal that half of the professional users prefer FB, closely followed by 45% of the users that prefer TW, while only 5% of the users choose G+ as their initial OSN for publishing their post. Furthermore, we compute the number of users that select first a particular OSN for more than 80% of their cross-posts, which shows a strong preference. There are 102 (16.56%), 75 (12.18%), and 5 (0.8%) users with a strong preference for TW, FB and G+, respectively. In summary, professional users are (more or less) equally divided into those that prefer TW and those that prefer FB, and very few cases that show a preference for G+.



OSN	#Users	%Users
FB	307	50
G+	30	5
TW	275	45

Table 13. Preferred OSN per user

2.3.3 Detection of Common Strategies by Sectors

The goal of this section is to verify the baseline hypothesis of whether the users of a particular sector present a similar behaviour in their use of OSNs. Then we first introduce the behavioural metrics used to describe the strategy of a user, and later apply the methodology proposed in [FVM13] to discriminate which categories follow our hypothesis.

2.3.3.1 Metrics to Capture Behaviour

The strategy of a user is defined by the decisions that he takes when posting information across several OSNs. Therefore, the elements we use to define the activity are behavioural metrics directly related to those decisions. Each behavioural metric is captured with one (or more) values in each OSN as it is detailed below. Overall each user is represented with a behavioural vector of 33 values that defines his strategy across FB, TW and G+. We wanted to provide the same weight to all the parameters, hence all the values range between 0 and 1 in the vector. This has led us to normalize one of the metrics, the activity rate. We have performed the normalization using the 90th-percentile of that parameter considering all the users in our dataset. All the users with a value above the 90th-percentile was assigned a value equal to 1 in the normalization. Note that we perform the normalization individually for each OSN.

Activity rate: We measure the average posts per day published by the user. As it is reported in [24], OSN users are intrinsically much more active in TW than in FB and G+. Therefore, we are interested on knowing how active is a user in a particular OSN with respect to the activity of other users in that OSN. With the proposed normalization for this metric we achieve that goal. This metric generates 3 values in the behavioural vector, one per OSN.

Fraction of Cross-Posting: We use as metric the portion of cross-posts in each OSN per user (3 values in the vector).

Cross-Posting pattern: We use as metric the portion of cross-posts happening in each possible OSN combination, i.e., FB-TW, FB-G+, TW-G+ or FB-TW-G+ (4 values in the vector).

Preference: This element is measured using the portion of cross-posts initiated in each OSN. This metric allows us to establish what is the preference of a user among the evaluated OSNs (3 values in the vector).

Type of content in regular-posts: This metric measures the portion of posts assigned to different types of content from the regular posts published by the user. In the case of FB and G+ the options are photo, video, link and text (only text or link in the case of TW). This metric generates 4 values in the vector for FB, one per type of content, 4 values in G+ and 2 Values in TW (10 values in total in the vector).

Type of content in cross-posts: This metric is similar to the previous one but in this case it only considers cross-posts (10 values in the vector).

2.3.3.2 Identifying Categories whose Users Present a Similar Strategy

We compare the similarity in the strategy of two different users by computing the Euclidean distance between their vectors. Hence, the lower the Euclidean distance the closer the strategies of the two users are. We can apply this process to compute what we refer to as intra-category and inter-category similarity. The former refers to the Euclidean distance between each pair of users within the category, while the latter is represented by the Euclidean distance of each user in the category to all the users outside that category.

We now apply the methodology proposed in [FVM13] to find the categories whose users present a similar strategy across FB, TW and G+. First, we measure the intra-category and inter-category cohesion of each category using a Kernel Density Estimation (KDE) [HTF01] method, where cohesion is measured based on the Euclidean distance. In addition, for each category, we run the Wilcoxon rank-sum test [WIL45] on the distributions of the intra-category and inter-category Euclidean distance. This is a non-parametric test of the null hypothesis that two populations are the same. The Wilcoxon test also provides the parameter W that measures the distance between the median of both distributions. In our analysis W equals Median inter minus Median intra, thus the larger W is the stronger is the intra-category cohesion. We note that we compute the parameter W as the difference of the medians in percentage (instead of absolute term) that provides clearer insights.

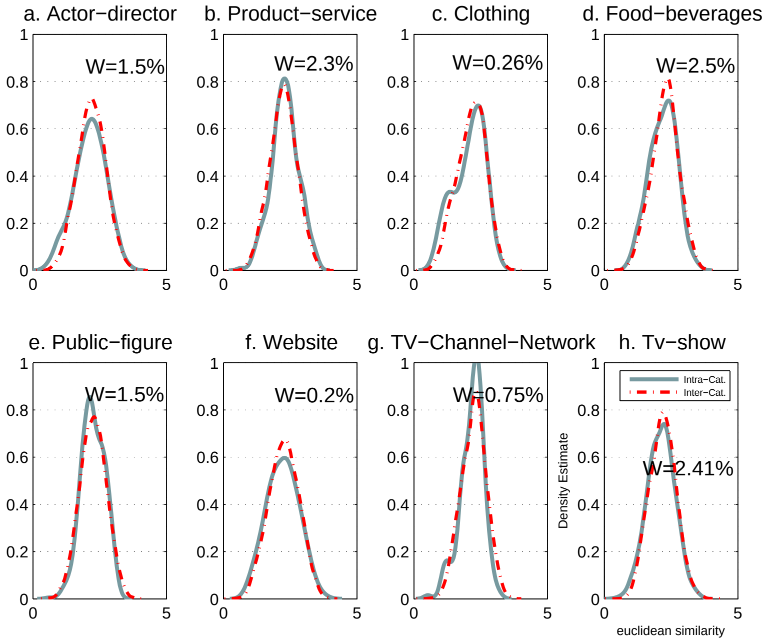


Figure 24. Kernel density estimation of the intra-category and inter-category Euclidean distance for those categories whose users do not present a common strategy

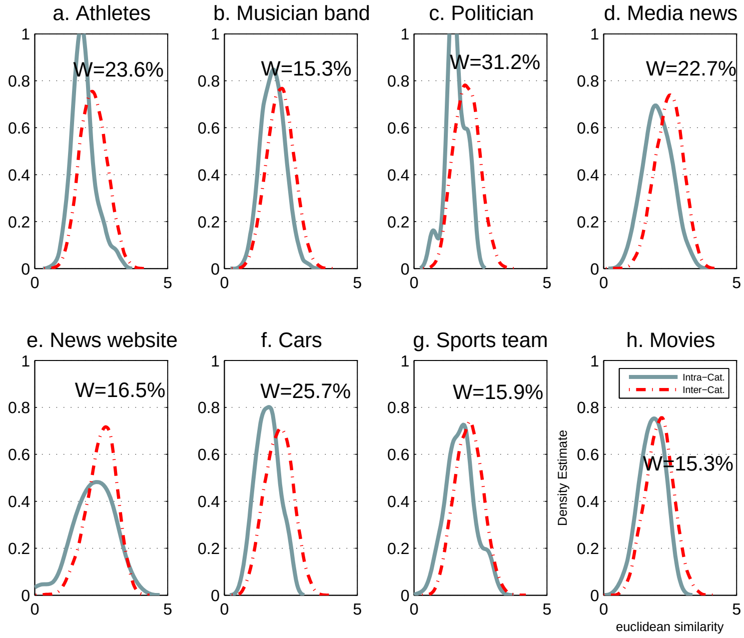


Figure 25. Kernel density estimation of the intra-category and inter-category Euclidean distance for those categories whose users present a common strategy

Figure 24 shows the KDE results for those categories in which the Euclidean distance among the users inside the category is very similar to the Euclidean distance with external users. This can be easily observed since the distributions are overlapped. Aligned to this result, the Wilcoxon test validates the null-hypothesis in all the cases (i.e., the distributions are the same), and W is below 2.5% in all the cases. Therefore, we conclude that the users in those eight categories do not present a common behaviour.

Contrary, Figure 25 depicts the KDE for those categories with a major intra-category cohesion. In this case, the Wilcoxon test rejects the null-hypothesis in all cases. This means that the intra-category and inter-category distributions are statistically different (p-value<0.001) for these eight categories. This statement is supported by the fact that for these categories W ranges between 15% and 30%. Therefore, these results uncover eight categories whose members present common behavioural elements (i.e., strategy) that globally differ from the strategy of the users outside that category. These eight categories are: Athletes, Cars, Media News, Movie, Musician-Band, News Website, Politician and Sport Team.

We note that from now on in the research the strategy of each category will be represented by the centroid of the category.

2.3.3.3 Similarity Between Categories' Behaviour

We have demonstrated that there are 8 categories whose users present a similar use of OSNs. However, the previous analysis neither says how close are the strategies of these categories nor defines the main elements of each strategy. In this subsection we address the first point, while the second question is covered in the next section.

To compare the strategies between two categories we calculate the Euclidean distance between their centroids. Figure 26 shows a colormap in which each cell unveils the Euclidean distance between the centroids of two categories. Visually, the closer the strategy of two categories is the darker the cell is.

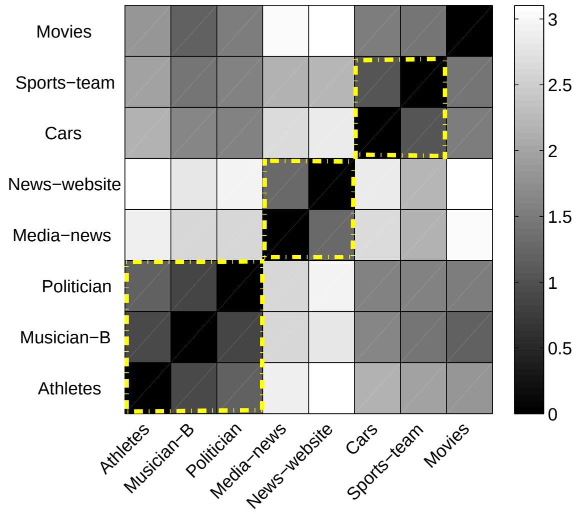


Figure 26. Colormap that represents the Euclidean distance between the behaviour of the eight categories with a similar strategy. The closer the strategy of two categories is the darker the cell representing their Euclidean distance. We find three relevant clusters among the analyzed users that are highlighted using a yellow dotted line.

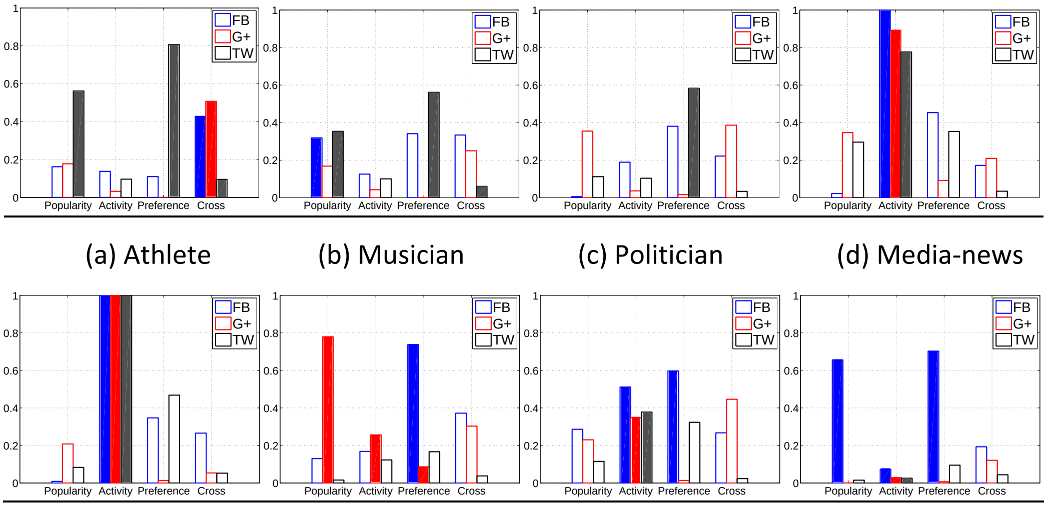
The results reveal three interesting clusters. First, Media News and News Website have very different strategies to any other category, while they present some commonalties in their use of OSNs. Second, the categories that represent individual users, i.e., Athletes, Music-Band and Politician, present a more similar strategy among them than to other categories. Third, Cars and Sport Teams, the two categories representing companies, present a major similarity to each other than to any other category. Finally, Movie present a strategy that is neither far away nor close to any other category except the two categories referring to news.

It is important to highlight that the fact that two categories present a higher similarity in their strategy does not mean they present exactly the same behaviour (i.e., the same values in the metrics). Instead, the correct interpretation is that those two categories will present some commonalities in some behavioural elements that make their strategies closer with respect to other categories.

2.3.4 Unveiling Strategies

In this section we reveal and discuss what are the most significant elements in the strategy of the 8 categories under analysis. Towards this end we use all the behavioural elements introduced above except cross-posting pattern because it is only relevant in the strategy of Cars. The other categories closely follow the general results reported before for this metric. In addition to the behavioural parameters, we use the popularity (i.e., number of followers) of each category in each OSN in the analysis. The reason is that although the popularity is not a behavioural element itself, it can influence the decisions of a user. As we did for the activity rate, we have normalized the popularity using the 90th-percentile in each OSN.

Figure 27 shows one bar plot per category in which each bar shows the value of popularity, activity rate, preference and fraction of cross-posts in each OSN, respectively. We have highlighted in full color the bars that represent the most significant elements of the behaviour of each category. In addition, Figure 28 shows the types of content in regular-posts and the types of content in cross-posts for each category and OSN, respectively. Following, we describe the strategy of each category:



(a) Athlete (b) Musician (c) Politician (d) Media-news

(e) News-website (f) Cars (g) Sports-team (h) Movie

Figure 27. Bar plot that shows the value of the following metric for each category and OSN: popularity, activity rate, preference and fraction of cross-posts

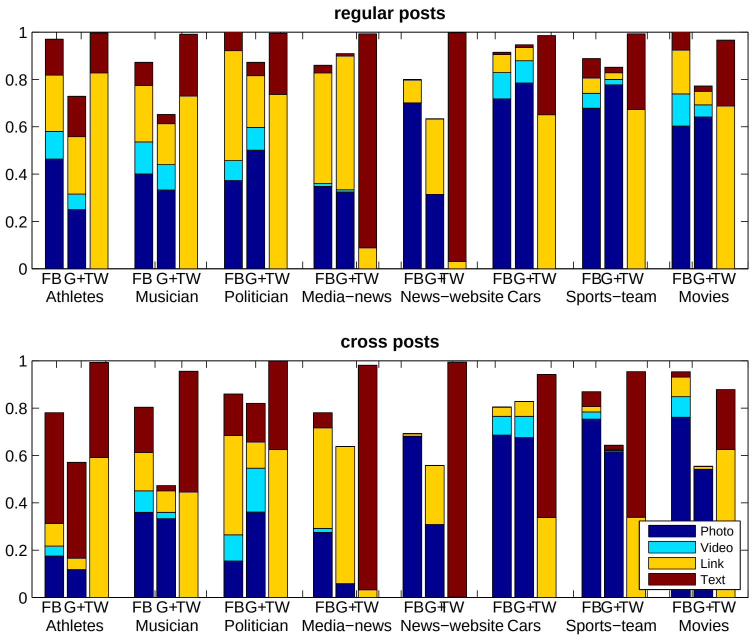


Figure 28. Bar plot that shows the types of content published in each category per OSN

Athlete: It is the category with the strongest preference for FB and with the most intense cross-posting activity in the three OSNs. It presents a low activity in all OSNs compared to other categories. Regular posts are mostly photos and links in FB and G+, however cross-posts are dominated by text in these two OSNs. This is explained because most of the cross-posts are initiated by TW (as shown by the strong TW preference) and replicated in FB and G+ as text. Finally, it is the most popular category in TW, which may explain its strong preference for this OSN.

Musician-Band: This category presents a clear preference for TW and an important level of cross-posting in this OSN (only surpassed by Athletes). The posts published in FB and G+ are mostly audiovisual content, both in cross-posts and regular-posts. The activity rate is low in the three OSNs. Finally, in terms of popularity, Musician-Band is the second most popular category in FB and TW behind Movie and Athlete, respectively.

Politician: Similar to Athlete and Musician-Band this category presents a preference for TW as well as a low activity in all 3 OSNs. The most interesting behavioural element of Politician is that it uses different content in FB and G+. Politician publishes more links in FB than in G+, where it mostly publishes audiovisual content. They also opt for using links in most of the tweets.

Media News: The differential strategy of this category is clearly a very high activity rate in the three OSNs. Actually, this seems reasonable since the users in this category are professional users (news agencies, portals, etc.) that are continuously publishing recent news. In addition, a second particularity of Media News is that the most common type of content in FB and G+ is link. However, it very rarely uses links in TW. In addition, together with News Website, this category shows a more balanced preference between FB and TW.

News Website: As the previous category, the differential behavioural element of News Website is its extraordinary high activity rate in all OSNs. In addition, News Media Website also shows a quite balanced preference between FB and TW. Contrary to Media News, in this case posts in FB are mostly photos, while in G+ they are balanced between photos and links.

Cars: Cars is the category with a major interest in G+, which may be due to its high popularity in that OSN. The behavioural elements that shows that interest are: (i) it is the only category in which the selection of G+ as initial source of information is relevant (it happens in almost 10% of the cross-posts), (ii) Cars is the only category in which its (relative) activity rate is higher in G+ than in TW and FB, and (iii) Cars is the only category in which the cross-posting activity between TW and G+ is not negligible since this pattern appears in 15% of the cross-posts. Apart from its interest in G+, Cars is clearly biased to FB in terms of preference and mostly uses audiovisual content in its posts. This seems reasonable since the business of Cars companies has a lot to do with presenting an attractive view of their cars and this requires the use of audiovisual material.

Sports Team: There are three elements that denote the behaviour of Sport Teams: first, a clear preference for FB, and second, an intense use of photos in its posts. Three, a considerably high activity in the three OSNs compared to the other categories (with the exception of the two categories related to news).

Movie: The behaviour of this category is defined by a strong preference of FB, the use of photos in most of its FB and G+ posts, and the lowest activity rate in the three OSN among the categories under analysis. This happens because the OSN accounts associated to movies are only active in a short period of time around their release and later they just keep a residual activity. Finally, there is a big contrast in its popularity since it is the most popular category in FB, but the least popular in TW and G+.

We conclude our analysis by enumerating the common behavioural aspects for the three clusters identified before. (1) All the individual users present a preference for TW and a relatively low activity in all OSNs compared to other categories. (2) Cars and Sports Teams, which represent commercial companies, shows a clear preference for FB and mostly post audiovisual content in FB and G+. (3) The categories related to news reporting coincide in having a very high activity rate.

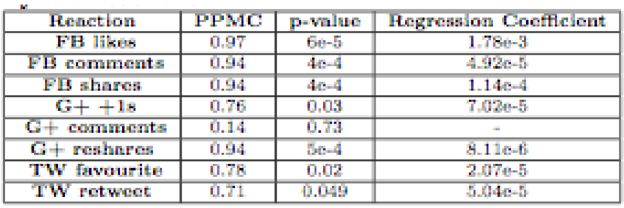
2.3.5 Evaluation of Strategies Success

To conclude this research we want to assess the success of the strategies adopted by the analyzed categories. To the best of our knowledge it does not exist any standard metric or methodology to evaluate the success of a strategy in OSNs. Our approach is based on the conviction that the number of reactions that a user attracts in his posts is the only objective available metric to capture the interest/engagement of end-users in the activity of a professional user. Therefore, in this research we propose to measure the success of the strategy of a category as a function of the average number of reactions that the category attracts per post. We believe that the proposed methodology is a useful tool to estimate the success of a particular strategy in the context of this section. However, we do not pretend to present it as a reference methodology to globally evaluate success in OSNs. Following, we first introduce our methodology and later we discuss the results extracted from applying it.

2.3.5.1 Methodology to Measure the Success Degree of Strategies

Our methodology proposes to compute the success of the strategy of a category as the difference between the expected number of reactions per post that category should receive and the actual number of reactions it receives. Therefore, our goal is to propose a model that estimates the expected volume of reactions per post for the eight categories under discussion.

Our intuition is that the number of reactions that a user attracts in a post in an OSN is strongly correlated to his popularity in that OSN. Therefore, our first step is to validate this hypothesis that would allow us to formulate the expected number of reactions as a function of the popularity.



Reaction	PPMC	p-value	Regression Coefficient
FB likes	0.97	6e-5	1.78e-3
FB comments	0.94	4e-4	4.92e-5
FB shares	0.94	4e-4	1.14e-4
G+ +1s	0.76	0.03	7.02e-5
G+ comments	0.14	0.73	-
G+ reshares	0.94	5e-4	8.11e-6
TW favourite	0.78	0.02	2.07e-5
TW retweet	0.71	0.049	5.04e-5

Table 14. Pearson coefficient, p-value, and Regression Coefficient of the correlation between popularity and reactions.

We calculate the Pearson Product-Moment Correlation Coefficient (PPMCC) between the popularity and all the reaction types separately. The PPMCC measures the degree of linear dependence between two variables, which becomes higher as the PPMCC moves to 1. Table 14 shows the PPMCC and p-value for the correlation associated to each reaction type. The results reveal a very strong linear positive correlation between popularity and volume of reactions per post for all types of reaction in all OSNs (PPMCC>0.7 and p-value<0.05). There is only one exception, G+ comments (p-value>0.05), which are omitted from our analysis in the rest of the section.

Based on these results, we propose a simple linear model that estimates the number of reactions a category should receive based on its popularity. Hence, we perform a linear regression to obtain the regression coefficient, listed in Table 5, associated to each type of reaction. In a nutshell, we estimate the number of reactions per post for a particular type of reaction in a category multiplying the popularity of that category by the regression coefficient for that reaction type.

Once we have the model to estimate the expected number of reactions we are able to evaluate the success of the different strategies. Figure 29 shows a colormap that represents the level of success of each category for each type of reaction. The colormap shows a positive (associated to green color) and negative (associated to red color) scale. For instance, a value of +2 implies that the category under analysis obtains twice as many reactions per post than what our model suggests. In contrast, a value of -2 indicates that the category is attracting half of the expected reactions per post. Note that the darker is the green color in a cell the higher is the success. Similarly, the darker is the red color in a cell the less efficient the strategy is. Each row corresponds to one category and presents a visual overview of the success of its strategy across the different OSNs and types of reactions.

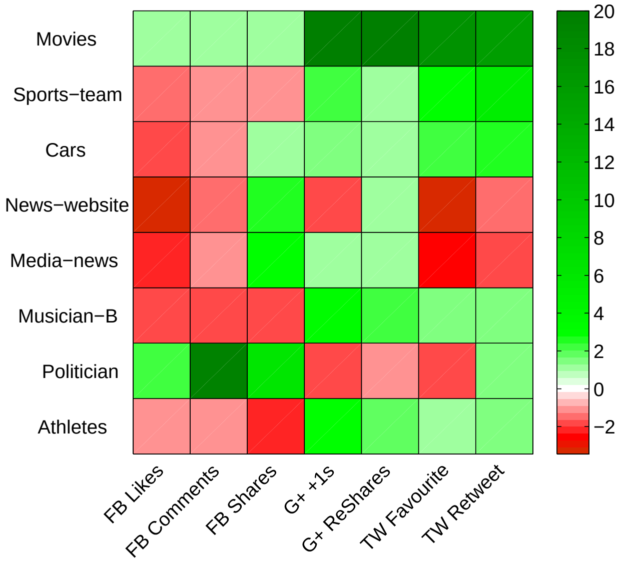


Figure 29. Colormap that represents the success of the strategy of each category across different types of reaction. The green color represents success and the red color represents failure.

2.3.5.2 Discussion of Strategies' Success

Movies is the only category with a successful strategy in all OSNs according to the volume of reactions it receives per post. This is an indicator that the adopted strategy is well adapted to the requirements of its audience in each OSN.

Athletes and Musician-Band are successful in TW and G+, but they fail in FB. Based on their clear preference for TW, it seems its strategy is adequate to cover their main objective, however they should modify their behaviour in FB in order to increase the engagement of end-users.

Politician has a successful strategy in FB, especially on attracting comments, but it fails in G+. In the case of TW it manages to get more retweets than expected, but does not cover the expectation in number of favourites. Its strategy is fair enough in FB to cover the expected reactions. In the case of TW, if its major interest focuses on spreading tweets its strategy is also adequate.

It seems that the interest of Cars in G+ is obtaining its reward since it manages to attract more reactions than the estimation of our model. In contrast, it seems Cars should revise their behaviour in FB since it only succeeds on the number of shares, even though it has a strong preference for this OSN.

Sports Team fails in FB, but is successful in TW and G+. Therefore, it should change some behavioural aspects to increase their engagement in FB.

Finally, Media News and News Website categories present a quite similar success pattern with the exception of G+ likes. We believe the most important types of reactions for news agencies and portals are share, reshare and retweet, since their goal is to spread the reported news as much as possible. For these reactions they present an almost identical result that reflects a success in FB and G+, but a failure in TW. This is a quite negative outcome since TW is considered a very relevant communication channel to disseminate news nowadays.

2.3.6 Conclusions

This research advances the state of the art regarding the strategy used by professional users in OSNs in three main elements. (i) To the best of our knowledge this is the first study that follows a data-driven approach to analyze the strategy of professional users in OSNs. (ii) We evaluate the global strategy of some professional sectors in the three major OSNs, namely FB, TW and G+. In contrast, most previous work focuses in the analysis of individual users and obtain adhoc conclusions. (iii) To the best of our knowledge, this research is the first one that proposes a quantitative estimation of the success of a strategy. In order to be able to make an analysis per sector, our first step has been to demonstrate that there are sectors whose users present similar behavioural elements that define a common strategy in OSNs. In particular, we have found eight sectors with a common strategy: Athletes, Cars, Media News, Movie, Musician-Band, News Website, Politician, and Sports Teams. The more interesting findings for the analyzed sectors are: (i) the two categories related to news show an extremely intense activity in the three OSNs; (ii) Athlete shows a strong preference for TW that directly impacts the information published in FB and G+; (iii) Cars gives a high value to G+ where they have a much stronger presence than any other category, and, (iv) Movie is very active around the release of the film but later the activity becomes residual. Finally, we estimate the success of each strategy. The success is measured as the difference between the actual volume of engagement (i.e., reactions per post) and the expected volume of engagement based on the popularity of the category. Movie is the only category that overpasses the engagement expectation in all OSNs. Politician is the only category, in addition to Movie, with a clear success in FB, but it is the only category that does not reach the expectation in G+. Finally, the news-related categories are the only ones that do not reach the expected engagement in TW, neither in retweets nor in favourites. In addition to all the previous findings, this work presents an aside contribution that characterizes the cross-posting phenomenon for professional users across FB, TW and G+. We have demonstrated that this phenomenon exists and is relevant. The dominant cross-posting pattern is FB-TW, while it is very rare to find information shared between TW and G+.

2.4 Analysis of BTLive Content Distribution

In deliverable D3.1 [D3.1], the first steps for the study of BitTorrent Live were presented. Since then, the measurements were continued and models for the system were derived to better understand the content distribution process of this new system. To this end, the following content was published as conference paper [RKH14] with the title "Clubbing with the Peers: A Measurement Study of BitTorrent Live" at the IEEE International Conference on Peer-to-Peer Computing 2014 in London and was awarded with the best paper award.

2.4.1 Introduction

The distribution of media content over the Internet has gained increasing attention over the last decades, resulting in a dominating part of worldwide network traffic [Cis13], [San13]. The increasing access bandwidth of consumers combined with the development of highly efficient video codecs, as well as new classes of end-user devices are inevitably changing the user behaviour in media consumption [Eri13] and have coined new application scenarios and business models for over-the-top video streaming. Examples include video-on-demand services like YouTube[2] for user generated content or Netflix[3] for movies and TV series. Even traditional TV broadcasters recently started providing their content as catch-up TV. In spite of these new services, studies show that linearly broadcasted TV content as well as live delivery of events still play an important role for the users [Eri12], [Eri13]. As a result, also more and more live content shifts to the Internet [Med13].

The distribution of live video content on an Internet scale has been extensively studied both in research and industry. While IP multicast would be a desirable and efficient approach for distributing live streams to a large number of users, it has a number of considerable drawbacks for network providers that could not completely be addressed in the last decades [DLL+00]. As a result, IP multicast is not available on an Internet scale and, in particular, not for the delivery of over-the-top traffic. Today, mostly cost-intensive centralized streaming systems are used, relying on individual IP unicast streams to clients. Due to the inherent limitations of this approach, content-delivery networks (CDNs) were deployed, such as the largest one by Akamai [NSS10]. With hundred thousands of servers all over the world, Akamai forms an overlay network on top of the Internet that is able to widely distribute content before it is delivered to the end users by nearby CDN nodes relying on unicast.

To further improve the distribution of content and to reduce costs for the content provider, a large number of decentralized, i.e. peer-to-peer (P2P) streaming approaches have been proposed [ZH12]. While some of them are meant to operate completely decentralized, also hybrid CDN-P2P approaches, such as Akamai's NetSession [ZCL+13] have been recently proposed and successfully applied. Using otherwise idle client resources i.e. upload bandwidth, P2P streaming systems are able to greatly reduce the load on the content providers as well as on CDNs. Especially for small content providers and live streams with hard to predict dynamics in the number of users, P2P streaming remains a promising approach. The key factors that are usually applied to investigate the streaming quality of such decentralized live streaming approaches include the achievable playback delay, i.e. the time between broadcasting at the streaming source and playback at the clients, as well as the

[2] https://www.youtube.com/ [Accessed July 30, 2014]

[3] https://www.netflix.com/ [Accessed July 30, 2014]

caused overhead. Depending on the delivery coordination and built topology, playback delays can vary between less than a second and minutes [ZH12].

In September 2012, Bram Cohen presented a novel P2P streaming system based on the so called screamer protocol [Coh12], specifically targeting live streaming with low delays and low overhead. The approach was filed as a US patent [Coh13] and published as a first implementation, called BitTorrent Live [4](BTLive), in March 2013. According to [Coh13], the specific benefits of the BTLive system are its low latency and low overhead. For content providers considering to use BTLive to broadcast potentially large-scale live events, it is essential to get an in-depth understanding of its properties and limitations. To the best of the authors' knowledge, there currently exists no publicly available study on BTLive's performance characteristics. For P2P streaming, in particular, the tradeoff between performance, in terms of delay imposed by the system, and the costs in terms of server load on the content provider itself, as well as the overhead imposed on the individual clients need to be well understood. As clients are contributing with their resources, they become a crucial part of the system.

To this end, this work presents a detailed measurement study of the BTLive streaming system conducted since March 2013. The system is currently down as the BTLive developers are preparing for a mobile version of the system. As basis for measurements, the official beta version of BTLive was used, which was accessible on the BitTorrent webpage until February 2014. The goal of this work is to quantify BTLive's key system characteristics and, thus, answer the following three key questions: (1) How P2P is BTLive? (2) How delay optimized is BTLive? (3) What is the overhead of BTLive?

2.4.2 BitTorrent Live

A huge variety of P2P live streaming protocols have been proposed over the years (see [ZH12] for an overview). In comparison to those systems, BTLive can be classified as hybrid streaming overlay that applies a number of different mechanisms at the same time. Figure 30 shows the structure of the stream delivery for an example configuration. This approach, which was introduced in [Coh12], could be confirmed during the measurement study presented in this work. The process conceptually consists of three stages: (1) a push injection of video blocks from the streaming source into the so called clubs, (2) an in-club controlled flooding, and (3) a push delivery to leaf peers outside the individual clubs.

The source divides the video stream into substreams, which is a well-known concept in multi-tree-based streaming [CDK+03], [LGL08]. A tracker is used for peer discovery, similar to the concept known from BitTorrent [Coh03]. The peers are divided into clubs, whereby each substream belongs to one of the clubs. The assignment to clubs is done by the tracker when a peer joins the system. A peer belongs to a fixed number of clubs in that it is an active contributor. In all other clubs, the peer is a leaf node and solely acts as downloader. The source plays a special role and always belongs to all clubs to inject the video stream to the individual clubs. Peers strive to establish multiple in-club download connections with members belonging to the same club as well as a single out-club download connection within each of the other clubs. To contribute to the distribution of blocks, peers can establish multiple upload connections to peers within the same club as well as to leaf nodes outside the club. The objective of a club is to spread video blocks of its respective substream as fast as possible to many nodes that can then help in the further distribution process. According to [Coh12], peers can be part of multiple clubs at the same time. This has the advantage of being able to balance upload resources across clubs, but is not required for the approach to work.

As mentioned, peers strive to establish a number of upload and download connections inside their own clubs to help in fastly spreading blocks of the respective substreams. In contrast to pure multi-

[4] http://live.bittorrent.com/ [Accessed July 30, 2014]

tree approaches, BTLive does not build up a single, stable datapath within the clubs. Instead, it builds a structure that can be classified as mesh topology, where peers may receive blocks within the club from any of the peers they have a download connection with. While typical mesh based streaming systems use a pull-based mechanism for a controlled block delivery process, BTLive relies on a pushbased mechanism, where peers push newly received blocks to all its upload connections within a club. This way, the costly exchange of blockmaps as well as the requesting of blocks, inherent to pull-based streaming, is avoided. While this implies a significant reduction of delays for the spreading of blocks, it also has a major drawback: it introduces the problem of duplicate block transfers. The reason is that the forwarding process is similar to a flooding of the mesh inside the club. While peers can locally avoid duplicate forwarding of the same block to the same neighbors, they cannot eliminate the high chance of delivering blocks that other peers concurrently send to the same neighbors. Due to the large size and number of video blocks, this can cause a significant overhead, reducing the overall efficiency of the streaming process.

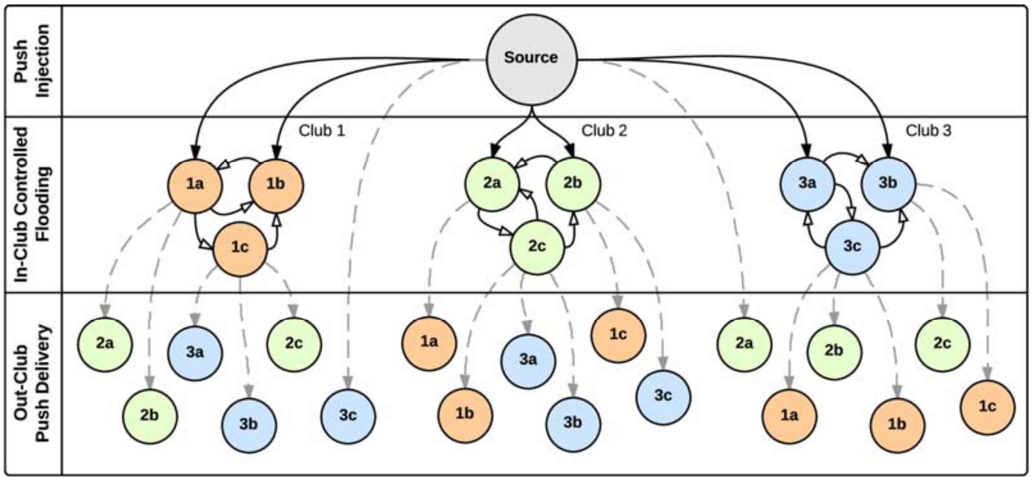


Figure 30: Three-stage delivery process of BTLive: (1) push injection of substreams into clubs; (2) in-club controlled flooding; (3) push delivery to peers outside the individual clubs

To mitigate this problem, BTLive includes an additional concept: Every time a peer receives a block within a club, it immediately sends out a message to all other download connections inside the same club to announce the arrival of the block to them. As this message is much smaller than a message including video payload, the chance of sending duplicates can be reduced as neighbors can quickly learn which blocks were already received by the peer and, thus, should not be pushed to the peer a second time. This reduces the chance of duplicates but cannot avoid them. Especially the fast spreading of new blocks and the inherent time dependency between blocks implies a high chance of peers within a club concurrently sending the same block at the same time.

In summary, BTLive's design strives to reduce the streaming delay at the cost of duplicate block transfers. In Section 2.4.4, this tradeoff is further discussed and the overhead caused by the beta version of BTLive is investigated under realistic conditions. The latter is important information to understand the performance and costs of the approach and to compare it to other state-of-the-art streaming solutions.

2.4.2.1 Theoretical Model

A simplified theoretical model has been derived to describe the processes of data distribution in BTLive and the bandwidth required at the streaming source, i.e. the broadcasting server. The model is derived from the protocol definition and, in particular, the delivery mechanisms that are tightly coupled to the concept of clubs. The purpose of this model is to allow for a worst-and best-case estimation of the required upload bandwidth at the source. Thus, it can help to describe when the

P2P effect can take over, meaning that new peers can completely be served by other peers, not affecting the bandwidth of the source.

For the model, in the following, it is assumed that each peer is a member of exactly one club, while the source is a member of all clubs. For simplification purposes, it is further assumed that all peers have sufficient upload bandwidth.

Two cases are distinguished: In the static case, connections remain active once established, while in the dynamic case connections with the source can be detached in the process to optimize the overlay structure. The former describes the worst case and the latter the best case in terms of required upload capacity at the source. Depending on the specific implementation of the protocol, overlay structures are expected to result in required capacities in-between these two cases.

1) Static case: The overlay structure resulting from the pure static case is depicted in Figure 31 on the left. In state 1, the first peer (A) joins the swarm5. It belongs to club 1 and requires 6 download connections with the source (S), one for each club. When the next peer (B) joins, the first peer provides it with the stream for club 1. The remaining streams are served by the source. The third peer (C) receives the streams for club 1 and 2 from the first and second peer and the remaining streams from the source. This case is called static as existing connections are not replaced by connections to other peers that join later. In this case, a peer is only served in full P2P fashion if at least one peer in each club already exists.

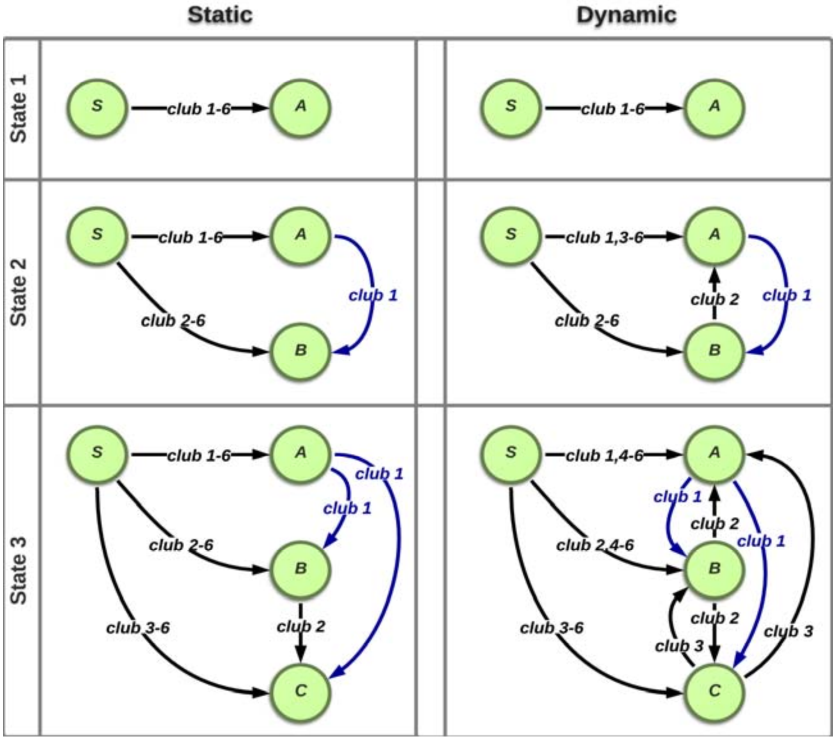


Figure 31: Comparison of the two cases for the first three joining peers

The minimum bandwidth requires at the source for the static case can be calculated using the following non-closed or closed-form expression:

[5] The term “swarm” refers to all active peers participating in the overlay.

$$bw_{\text{static}} = b_{\text{trans}} \times \sum_{i=0}^{c-1}(1-\frac{i}{c}) = b_{\text{trans}} \times \frac{c+1}{2}$$

where c is the total number of clubs and btrans the bitrate required for transmitting the video stream using the BTLive message format. The latter can be calculated by adding the BTLive protocol overhead to the average video bitrate.

For 6 clubs, bw_{static} would result in 3.5 times the transmission bitrate btrans. This factor increases to 6.5 for 12 clubs. The ratio of the number of clubs to the required resources in the static case is illustrated in Figure 32. As expected, it shows that the required source upload bitrate increases steadily until there is one member for each club. Subsequently, it levels off and the peers are fully providing the streams for all additional peers that join thereafter.

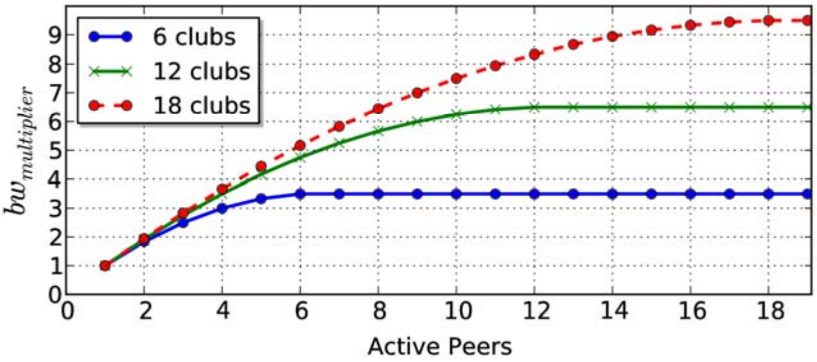


Figure 32: Required source upload bandwidth in the static case. The absolute value is calculated by multiplying this factor by the transmission bandwidth.

2) Dynamic case: This case is depicted in Figure 31 on the right. A suitable replacement of connections results in less bandwidth required at the source. Thereby, it is the most ideal process for minimizing required source bandwidth. In state 1, (A) joins and subsequently gets all clubs from (S). In this state, there is no difference between the two cases. The difference comes into effect in state 2, where (A) terminates its connection for club 2 and establishes the same connection with (B) instead, which is member of club 2 and, thus, can help to offload the source.

In the dynamic case, the minimum bandwidth required at the source can be calculated as follows:

$$bw_{\text{dynamic}} = b_{\text{trans}} \times \sum_{i=0}^{\lceil \frac{c}{2} \rceil - 1} (1-\frac{2i}{c})$$

For brevity reasons, the closed-form variant is omitted here.

For 6 clubs, bw_{dynamic} results in twice the transmission bitrate btrans. The factor increases to 3.5 for 12 clubs. The ratio of the number of clubs to the required resources in the dynamic case is illustrated in Figure 33. It shows that the required source upload bitrate increases steadily until there is one member for c/2 clubs. Subsequently, the peers start to release the source and the required source upload bitrate decreases until there is one member for each club. It then remains at a level of the transmission bitrate btrans and the P2P effect completely takes over all remaining upload.

The theoretical model shows that the minimum number of peers necessary for the P2P effect to start as well as the minimum upload bandwidth required at the source both strongly depend on the number of clubs. The higher the total number of clubs, the more upload bandwidth is to be provided by the source and the more peers are required for the P2P effect to take over. In the remainder of the document, this inherent limitation of BTLive is referred to as source bottleneck problem. Following the theoretical model, this problem leads to increased server bandwidth requirements in scenarios where the number of peers is smaller than the number of clubs. As the number of clubs is a fixed configuration parameter of the BTLive system, it cannot easily be changed during runtime. Furthermore, having a minimum number of clubs is essential to maintain a multi-tree topology with all its benefits [LGL08]. Thus, an adequate choice for this parameter is important and can be highly dependent on the target scenario.

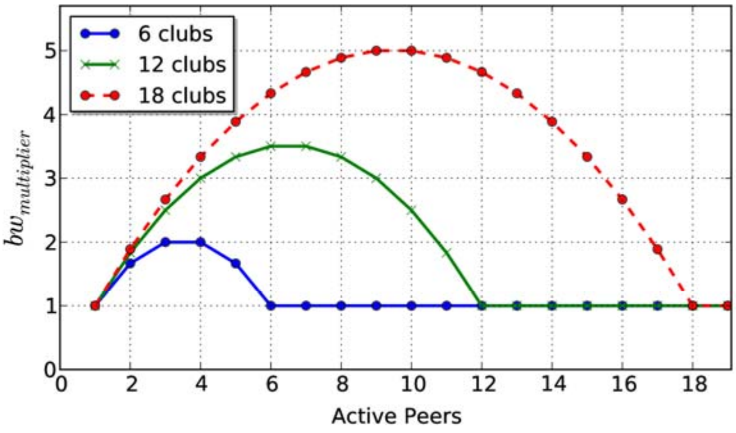


Figure 33: Required source upload bandwidth in the dynamic case. The absolute value is calculated by multiplying this factor by the transmission bandwidth.

2.4.3 Measurement Methodology

The Emanicslab testbed[6] with 20 nodes distributed all over Europe was used to run the streaming source and the clients during the study. Depending on the number of peers involved in the individual measurements, all available sites of the testbed were used. For experiments exceeding the number of physical nodes, multiple peers were run at individual machines within the testbed. The tracker software was not published and, thus, could only be run by BitTorrent Inc. Based on the observed IP address during the study, a single tracker seemed to be run most likely using Amazon EC2 web services[7].

The network traffic at the individual clients as well as the source was captured and studied to derive statistics related to individual peer connections, different packet types, and the protocol flow. Besides, timing-related measurements were conducted to study the streaming delay properties of BTLive. Hereby, the streaming delay is defined as the difference in time between the source sending out a media block and an individual client receiving it. This can be a block that was forwarded directly in only one hop from the source to the client or a block that was propagated throughout the streaming overlay over several hops. Therefore, it describes how long a client has to wait until content is available at its side and can be handed over to the media player for playback. The actual playback delay can be longer, depending on the decoding process and the playback state of the video player. As the used player software could not further be investigated and the playback and the buffering strategies are usually out of control of the streaming overlay, it is not considered here.

[6] http://www.emanicslab.org/

[7] https://aws.amazon.com/de/ec2/

2.4.3.1 Network Traffic Capturing and Analysis

Figure 34 schematically shows the measurement setup across different entities used as vantage points in the study. The streaming software was provided by BitTorrent Inc. as binary that is to be installed by the user and includes the streaming client only. For video playback, the Adobe Flash Player is used, communicating with the local streaming client using RTMP. A client usually visits a channel-specific webpage, where the Flash-based video player is embedded, initializing the connection to the local streaming client and passing it the required information to connect to the channel's swarm. The information includes a channel identifier as well as the contact information of the tracker, which are embedded in the webpage. To be able to run the client software on the headless nodes of the testbed, the tool RTMPdump was used to emulate a video player that is connected to the streaming client and to dump the received video stream to a file. This was done to be able to check the playback quality of the individual clients.

TCPdump[8] was used as measurement tool to capture traces of the incoming and outgoing BTLive network traffic as well as the local RTMP traffic for later analysis. The BTLive traffic is further analyzed using a Wireshark plugin[9], which was developed for this purpose[10]. The RTMP traffic is used to determine the streaming delay. Moreover, the payload of the BTLive traffic needed to be analyzed in detail, as the BTLive protocol structure was not published at the time of this study. Nevertheless, the most important parts of the packet format were identified by using the available information from [Coh12] and an in-depth study of recorded network traffic traces. This way, the role of the different message types and relevant fields of these messages could be decoded.

2.4.3.2 Measuring Streaming Delay

For the streaming delay measurement, first, the clock differences of the nodes were determined using NTP at the beginning of each evaluation run. Using the RTMP traffic traces of the streaming source as well as the individual clients, different methods were investigated to determine the streaming delay. It turned out that the payload of the captured audio packets allows for a reliable identification of individual data blocks within the media stream. As the audio is played back synchronously with the video, this showed to be a good way to study the delay of the overall streaming process in a very finegranular manner. Therefore, the RTMP traffic was processed using a custom-built software tool, comparing and matching the individual audio packets sent out by the source and the ones received at the individual clients. Using the time difference information between the machines and the timestamps captured within the TCPdump[11] traces, the delay could be calculated for each individual packet. To make sure that matching of audio packets works correctly and does not falsely match duplicates in the stream, duplicates were recorded and excluded during the processing of the data. Duplicates can occur, e.g., if a short video is broadcasted in a loop or a video includes exactly the same audio sample multiple times. After identifying this problem and using a video longer than the duration of the evaluation runs, duplicates were avoided completely. As presented in the evaluation, using this method, in average between 10,000 and 12,000 matching samples per client in a 5-minute streaming session were recorded, allowing a detailed and accurate study of the delay characteristics.

[8] http://rtmpdump.mplayerhq.hu/

[9] http://www.wireshark.org/

[10] http://www.ps.tu-darmstadt.de/research/btlive

[11] http://www.tcpdump.org/

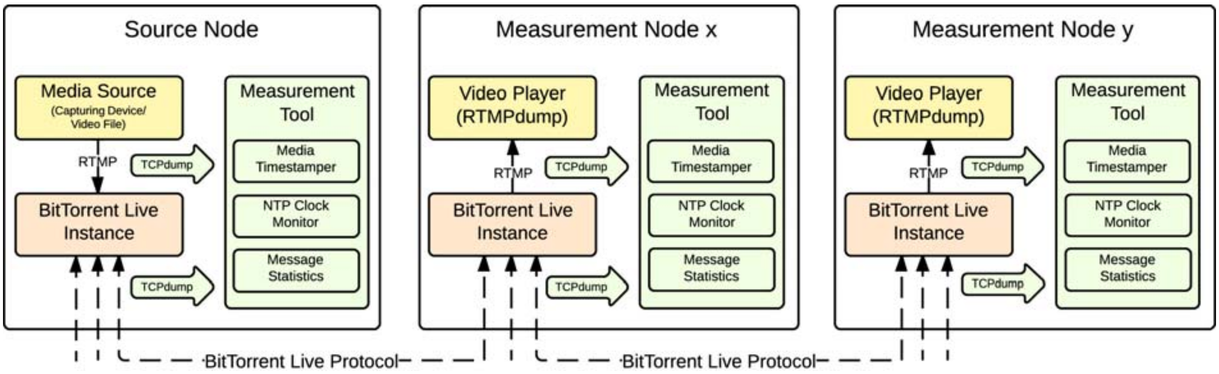


Figure 34: The setup of measurement tools across the different vantage points of the measurement: the streaming source and the measurement nodes.

2.4.4 Measurement Results

All experiments were conducted using an own channel, where a test video was broadcasted by a streaming source running on a dedicated host. The test video[12] was encoded with an average bitrate of 471 kbit/s. Depending on the experiment, different numbers of peers were run on Emanicslab machines. For the measurements, the latest published version (0.4.12.335) of the BTLive client was used. Using the Linux TC (traffic control) tool, the upload bandwidth of the source was limited to 4 times the average video bitrate, as recommended by BitTorrent Inc. as minimal upload bandwidth for the source. Experiments were repeated 10 times and 95% confidence intervals are reported for all averaged values.

2.4.4.1 How P2P is BTLive?

Even if P2P live streaming cannot work without a streaming source, the goal of any P2P approach is to shift load away from the source to the peers, i.e. the amount of data delivered by the source should be small in comparison to the peers. To understand how well BTLive achieves this goal, different system configurations with respect to the swarm size were studied, comparing their traffic characteristics. In all cases, the traffic measurements were conducted at a single peer that enters the system after all other peers already joined the swarm. This way, the start-up phase of bootstrapping the overlay is skipped as the goal was to study the streaming process after the system stabilized. The overlay traffic was recorded at the measurement peer for five minutes, after which the peer left the system. The five different swarm sizes studied are labelled according the number of peers present in the swarm, excluding the source and the measurement peer itself: 0, 2, 10, 50, and 90. As mentioned before, for scenarios with more than 20 peers, multiple peers were run on the individual testbed machines.

First of all, the number of clubs used by the overlay was studied based on a sample BTLive traffic trace. The trace revealed that in the observed BTLive version the number of clubs is configured to be 6, a number much smaller than expected, since from [Coh12] it seemed that a number of 12 clubs would be a desired configuration. Besides, the traces showed that, as expected, the source is an active contributor to all of the 6 clubs as it initially injects the data into the clubs. Furthermore, it turned out that the peers exclusively belong to only one club, which reveals that the advertised load balancing mechanism [Coh12] to efficiently distribute heterogeneous peer resources among clubs is not implemented so far.

In the context of the previous finding, the observed reduction in the number of clubs could be explained by the source bottleneck problem that was identified in Section 2.4.2. A study of the

[12] http://www.bigbuckbunny.org/

channel popularity of the beta version of BTLive over several months, which was conducted in parallel to this measurement study, showed that often most channels were not used at all or only by a small number of clients. While for large-scale streaming scenarios a higher number of clubs is desirable [Coh13], for scenarios where the number of peers is most of the time smaller than the number of clubs, the source bottleneck problem would clearly hinder the P2P effect and force the source to handle most load of the system. This could explain why the developers configured the protocol to use only 6 clubs. Furthermore, this implies that the number of clubs has to be chosen carefully, depending on the scenario.

Building up on these findings, Figure 35 shows the relation between video data served by the source and peers, for the five different swarm sizes as described above. Figure 35 and Figure 36 depict the data volume and total number of BTLive packets, respectively, sent and received by the measurement peer and averaged over 10 repetitions of the measurement. The peer received on average roughly the same amount of data and packets for all five swarm sizes. As expected, for a swarm size of 0 and 2, the measurement peer has no or only limited chances to contribute to the video packet distribution. Almost all packets are coming from the source. With no other peers in the channel, the measurement peer on average received 20.1 MB of video packets (17.1 thousand) and did not sent any video packet. It further received 0.5 MB (9.1 thousand) control packets from the source and sent 1.4 MB (24.2 thousand) control packets to the source. The average size of the video that was received and saved by RTMPdump was 16.4 MB. The difference between the video data and the volume of received video packets of around 3.7 MB (18.4%) is assumed to be caused by message overhead (i.e. roughly 5% for video packet headers) and the transport format used for video data transmission. The latter is supported by the observation that video data was transformed by RTMPdump after receiving and before storing it for later investigation.

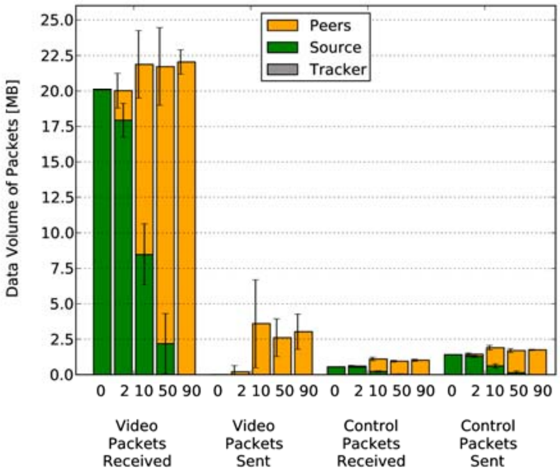


Figure 35: Data volume of BTLive packets at measurement peer for different swarm sizes

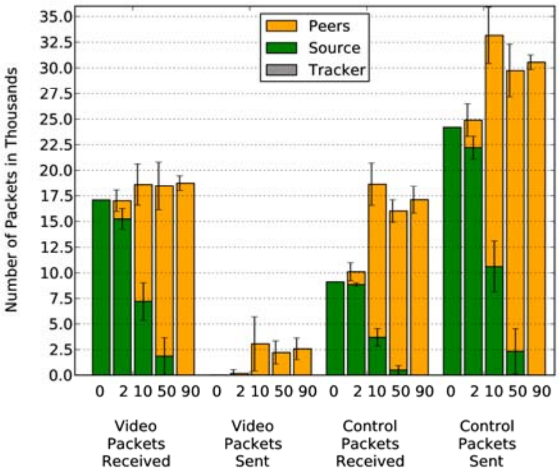


Figure 36: Number of BTLive packets at measurement peer for different swarm sizes

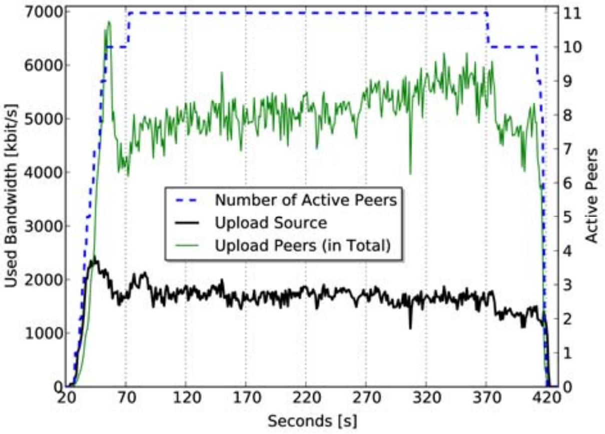


Figure 37: Comparison of upload bandwidth contributed by source and peers over time for the complete streaming session averaged over 10 measurements

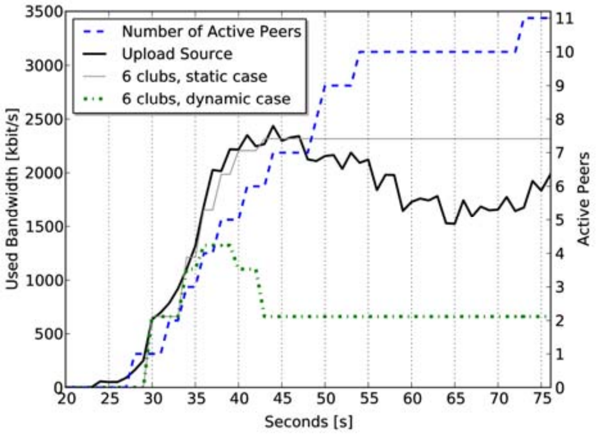


Figure 38: Upload bandwidth contributed by the source within first 75 seconds averaged over 10 measurements

In all settings, the data volume for sending and receiving control packets was below 3 MB, which roughly refers to less than 11% of the overall sent and received data. For the amount of received video data, the expected P2P effect is clearly visible with an increasing swarm size. If the peer is the only peer in the swarm, it naturally receives only packets from the source. With 2 other peers in the swarm, on average about 11% of the video data is delivered by peers. With 10 other peers, the share already increases to about 61% in average served by peers. With 50 peers, on average about 90% are served by peers, while with 90 peers, the measurement peer did not receive any video data from the source. The communication with the tracker is negligible and thus not visible in the figures. For the number of received and sent video packets, the trends look similar. A major difference is visible for the control packets. While they only make up for less than 11% of the data traffic, in all cases more control packets are sent than video packets are received. Besides, looking at the confidence intervals, the number of received control packets reaches a level that is not significantly different to the number of received video packets. This rather high number of control packets is assumed to be a result of the screamer protocol [Coh13] where the arrival of video packets is announced to all in-club download connections to avoid duplicate video packet transmissions.

Figure 37 provides an alternative view on the streaming process as it shows the contribution of upload bandwidth by the source and the peers over time for a swarm size of 10 peers (plus the measurement peer), averaged over the 10 repetitions. Here, the traffic was captured and aggregated at all peers to provide a swarm-wide view on the used resources. Figure 38 shows the upload bandwidth used at the source for only the first 75 seconds of the measurement. Both figures show that the average upload rate of the source increases by each newly joining peer until it reaches its maximum of 2,436 kbit/s after 44 seconds, i.e. after the 7th peer joined the channel. Subsequently, the upload rate of the source decreases, even though 3 more peers join the swarm. At the same time, the upload rate of the active peers is increasing, especially directly after peers number 8 and 9 join. It reaches its maximum at 6,816 kbit/s after 56 seconds, with 10 peers in the channel. The peak in the contributed upload bandwidth at the source confirms the theoretical model in Section 2.4.2, according to which for 6 clubs the minimum upload bandwidth is between 2 (dynamic case) and 3.5 (static case) times the transmission bitrate, which on average was about 662 kbit/s in this case.

Both the dynamic and the static cases are also depicted in Figure 38. The comparison of the actually observed source bandwidth with the two theoretically derived curves shows that the BTLive beta version closely follows the worst (static) case until second 46. Afterwards, it slightly decreases, which is also visible in Figure 38 for the rest of the measurement duration. This shows that connections by the source seem to be only occasionally replaced by connections from other peers. Thus, the P2P effect could be easily improved if the source would more aggressively close connections to clubs that are served by multiple connections, forcing peers to take over more load. If this should actually be done, is up to the content provider as surplus source bandwidth could also help to cope with peer churn and temporal inefficiencies in the streaming process.

Furthermore, the upload bitrates in an exemplary period of the steady state (i.e. between seconds 310 and 369) were determined. This period of time is considered steady as it lies between the initial decline on the source and the first leave of a peer. Any other period between about seconds 70 and 370 could have been used here. In this state, with 11 peers in the swarm, the total upload bitrate was 7,285 kbit/s, which on average corresponds to a 662 kbit download bitrate for each of the 11 peers in the channel. The average upload bitrate of the source in steady state was 1,648 kbit/s (22.62% of the load), while the average total upload bitrate of all peers in steady state was 5,637 kbit/s (77.38%).

In a recent interview [Dre13], an employee of BitTorrent Inc. stated that the P2P effect takes over at around 10 to 20 concurrent viewers and that from there on it scales with a growing number of users with no extra cost at the source. Indeed, the measurement results showed that the peers take over more load after a minimum number of 6 peers are present to help spreading the blocks of the stream to all 6 clubs. Before that, the identified source bottleneck problem hinders the P2P effect to take over. If peers do not have enough upload bandwidth to forward the blocks of their club, clearly the effect takes more peers to take over. Yet, it was also shown that the P2P effect could even be improved by more aggressively reducing contributions by the source.

2.4.4.2 How Delay Optimized is BTLive?

BitTorrent Inc. recommended a source upload bandwidth of at least four times the average video bitrate. Thus, for the delay study, an experiment was conducted where the source upload bandwidth was limited to 2,000 kbit/s (roughly 4×471 kbit/s). Especially for configurations with low peer bandwidths, this setting showed to be too low to maintain a stable streaming process. This can be explained using the model presented in Section 2.4.2. Following the worst (static) case of the model and the observed parameters of BTLive, the source should be configured to provide at least 3.5 times the transmission bandwidth (662 kbit/s), resulting in 2,317 kbit/s. Therefore, a second experiment was conducted with an increased source bandwidth of 2,400 kbit/s, which showed a stable streaming behaviour across the measurement peers. This implies that the recommendation for the minimum bandwidth should be higher.

To investigate BTLive's delay characteristics, a number of experiments with changing peer bandwidth configurations were conducted. For realistic network delays, 10 peers were deployed on dedicated machines at distinct Emanicslab sites. The observed average streaming delays for the different configurations are depicted in Figure 39. Additionally, the figure includes the number of matched samples that were used to calculate the average delays. The latter was added to verify that a sufficient number of samples was matched to draw valid conclusions. Although the number is slightly lower for the measurement with a low source and peer bandwidth, the total number of 10,000 samples is still on a high level.

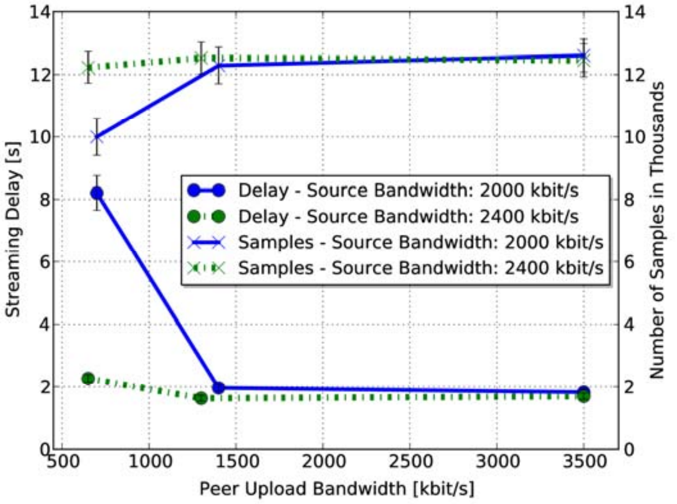


Figure 39: Average measured streaming delay and number of matched samples

For the first measurement series with a reduced source upload bandwidth (2,000 kbit/s), an interesting effect can be observed when also limiting the peers to a low upload bandwidth (700 kbit/s). The average delay (around 8 seconds) was significantly higher than for all other configurations. At the same time, a drop in the number of matched samples by around 2,000 samples was observed. An in-depth analysis of the captured data showed that the collected traces were significantly smaller than the ones of the other measurements, reducing also the number matching samples. Both can be explained by peers either not receiving the complete stream due to the scarce resources, or by peers leaving the swarm due to bad performance. By increasing the upload bandwidth at the source to 2,400 kbit/s, the average streaming delay significantly dropped to around 2 seconds, although the peer's upload bandwidth was slightly decreased to 650 kbit/s in the second measurement series. For all other configurations, the average streaming delay was observed to be below 2 seconds.

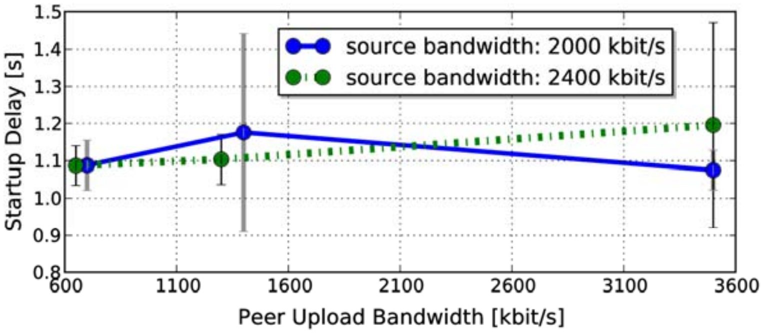


Figure 40: Average startup delay over the different configurations

Besides the average streaming delay, another important aspect is the startup delay as it greatly influences the abandonment rate of users in video streaming systems [Sit13]. Startup delay was defined as the time between a peer sending a first packet (i.e. the join message to tracker) and the arrival of the first media packet. This definition only captures the overlay characteristics. The real startup delay observed by the users is higher and depends on the video player, its buffer management, and the implemented playback policy. Figure 40 shows that the observed average startup delay is very low and stable, ranging between roughly 0.6 and 1.2 seconds for 95% of the measurements across all configurations. However, in all configurations single outliers were observed were the startup delay was significantly higher, ranging up to 15.5 seconds. To study this in more detail, Figure 41 shows the distribution of the measured delays over all samples. Here, the difference between the configuration with scarce source upload capacities becomes clearly visible. While for all other cases, the delays spread between 0.7 and 4.6 seconds, for this configuration, it varies much more and spreads from roughly 2 to 17 seconds. This clearly shows that the streaming process was suffering due to insufficient resources at the source, implying that the minimum capacity specified by BitTorrent Inc. is too low for cases were peers have low capacities as well. Some more measurements were conducted in which the source bandwidth was set to even lower values than 2,000 kbit/s, resulting in highly unstable streaming processes. This observation initially led to the theoretical considerations presented in Section 2.4.2, providing a good explanation of the system behaviour.

According to BitTorrent Inc. [Dre13], the streaming delay averages around five seconds, regardless of the swarm size. With 10 peers in the channel, the lowest average latency observed was 1.631 seconds in the case of 2,400 kbit/s source and 1,300 kbit/s peer upload bandwidth. In sum, the low delay properties of BTLive could be confirmed in the measurement study for small swarm sizes, as long as the source capacity was higher than the theoretically derived threshold.

2.4.4.3 What is the Overhead of BTLive?

Overhead in video streaming is mainly a result of control traffic or duplicate transmissions due to lack of coordination.

For P2P overlay maintenance and data exchange coordination, control traffic is inevitable and can greatly differ depending on the applied topology and delivery concepts. In comparison to the delivered media data, the volume of control traffic is usually negligible. Nevertheless, the large number of control messages can cause a significant load on the network.

Besides control traffic, duplicate packets can significantly contribute to overhead. As described earlier, due to its aim to reduce streaming delay, BTLive follows a controlled flooding approach in which peers send out announcements of packet arrivals to other in-club peers as soon as a packet was received. This approach accepts that media packets may actually be delivered multiple times to the same peer. To show that this is actually the case, Figure 42 depicts the data volume caused by both control traffic and duplicate media packets. For the same configurations as here, Figure 35 and Figure 36 above showed how the data volume compares to the number of packets. It is notable that the increase in volume and number of video packets due to duplicate video packets is very well visible for 10, 50, and 90 peers. Distinguishing further between unique and duplicate video packets, unique video packets of about 20 MB on average per peer were observed with only small variation across all measurements. In contrast, the data volume caused by duplicate media packets shows more variation as the steep increase between a swarm size of 2 and 10 peers shows. It is apparent in Figure 42 that for a swarm size of 10 peers or higher, the overhead from duplicates ranges between 1.7 and 1.9 MB, relating to roughly 8% of the video data. At the same time, the overhead from control traffic ranges between 0.5 and 1.1 MB. In sum, the overhead, including both duplicates and control traffic, on average accounts for roughly 12% of the overall traffic, where around 65% of the overhead is caused by duplicates. Compared to other state-of-the-art systems this overhead is very high. In [WRR+14] it was shown that typical mesh and tree-based approaches exhibit roughly between 0.5 and 5.5% overhead and that a hybrid approach can achieve less than 1% overhead, even for challenging scenarios. This clearly shows the price BTLive pays for low streaming delays.

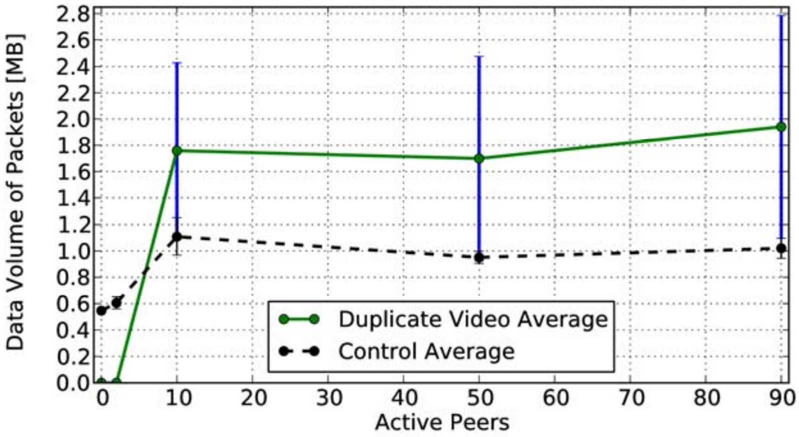


Figure 41: Average data volume of control traffic and duplicate video packets observed at the measurement peer in scenarios with different numbers of peers

To further study the overhead, the traffic caused inside and outside clubs was distinguished. Figure 43 shows the different classes of overhead and, for comparison, the unique video packets delivered. The result supports the earlier findings for the cause of overheads. For each club that a peer is not a member of, the peer has one out-club download connection.

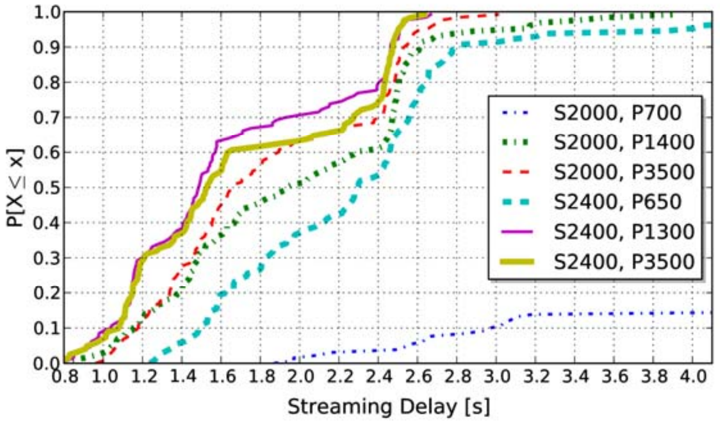


Figure 42: Distribution of streaming delay

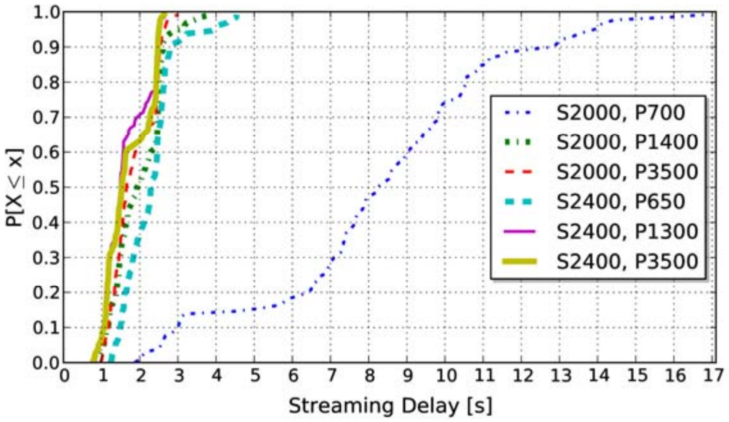


Figure 43: Distribution of streaming delay (zoomed in)

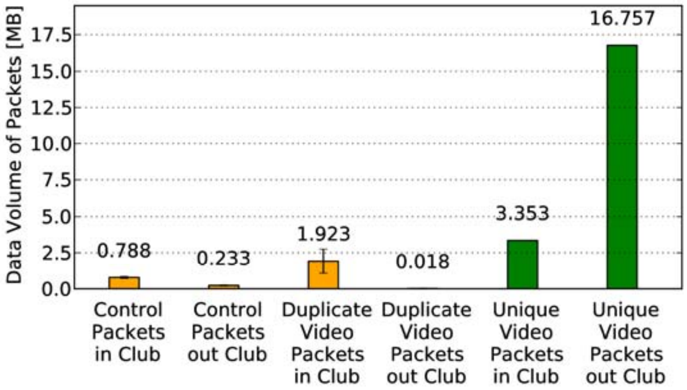


Figure 44: Average data volume for different traffic classes, inside and outside clubs captured at the measurement peer in the scenario with 90 peers

Using these connections, most of the unique media packets are delivered, which on average account for 16.8 MB, while almost no duplicates were observed. For the club to which the peer is an active contributor, it was observed that it has between 2 and 3 in-club download connections. Here, it is possible that multiple in-club uploaders send the same media packet at the same time, causing a duplicate transmission for the receiving peer. Therefore, both unique and duplicate video transmissions were observed for the in-club case. On average, about 3.4 MB of unique and 1.9 MB of duplicate media packets occurred. The latter translates to roughly 8.6% of all received media packets. CDFs are not shown due to space restrictions. They showed very stable results for all traffic classes except for the duplicate in-club packets, which varied between 0.06 and 3.29 MB. This means that in some cases, the data volume caused by in-club duplicates were as high as for the unique media packets, even though around 0.79 MB of control packets were received inside the club, mainly caused by packets that announce the arrival of new media blocks. This is observable in Figure 36, which shows that the controlled flooding approach inside the mesh-based clubs imposes significant overhead, compared to the media data transmitted within the club. Out-club deliveries happen without any duplicates, as expected for a pure push-delivery over a tree topology.

In sum, while BTLive avoids exchanging block maps in the mesh-based clubs, it compensates for that by sending announcement packets for the arrival of blocks, resulting in a large number of control packets being sent. In spite of this delivery coordination, BTLive causes significant overhead in form of duplicate video packets that are avoided by other approaches not combining mesh-and push-based delivery.

2.4.5 Related Work

To the best of the authors' knowledge, there currently exists no publicly available measurement study for BTLive. However, various studies have been conducted over the last decade analyzing other P2P streaming systems. Most of these studies analyzed PPLive or PPStream.

Vu et al. [VGL+07] and Hei et al. [HLL+13] conducted active measurements to study overlay and streaming session characteristics of PPLive. The first work studied graph properties of the overlay and, e.g., shows that PPLive overlays up to a certain size can be described as random graphs and that the average peer degree is independent of channel populations. The second work shows that PPLive causes long start-up and playback delays, ranging from several seconds to a couple of minutes.

Liang et al. [LWB+09], in contrast, conducted passive measurements of PPStream during the 29th Beijing Olympics to study streaming characteristics. The authors show that the Olympics' channels differed from the rest of the channels in terms of playback delay and smaller scheduling units to achieve a timelier and more reliable, yet less efficient delivery of the streams. Another passive measurement study was conducted by Gao et al. [GLx13], showing that both systems PPLive and PPStream can provide an excellent viewing experience for popular channels but are rather inefficient for unpopular channels. Alessandria et al. [AGL09] conducted a passive measurement study for different commercial streaming systems, namely PPLive, SOPCast, TVants, and TVUPlayer. The authors observed that all applications avoid bad network paths and carefully select neighbors. Furthermore, they found that the behaviour of the peers can get aggressive if there is a bottleneck which affects all peers, for example at the access link. A more recent study from 2013 on SopCast by Vieira et al. [VSH+13] observed that SopCast channels tend to have users in the order of hundreds but can become much larger during special events. Their measurements show that almost 75% of all packets exchanged between peers were control messages.

Some of the studies discussed above used a similar approach as the one applied here for measuring a given P2P streaming system under realistic conditions. Moreover, this work studied BTLive in a controlled environment, i.e. the source and all peers in the study were under the authors' control. This allowed for an in-depth understanding of the streaming process primitives of this novel streaming protocol.

2.4.6 Discussion and Conclusion

As stated in the beginning, the goal of this work was to answer three key questions related to the characteristics of BTLive. The first question was: how P2P is BTLive? This work shows that the P2P effect takes over with an increasing number of peers in the swarm, above a minimum number of 6 peers (equal to the number of observed clubs). In addition, the used bandwidth at both source and peers was studied over time. The required upload resources at the source were described by a theoretical model, matching the observed source bottleneck with a small number of peers. The model includes both an estimate for the lower and the upper limit for the required upload bandwidth at the source over time. The results show that there exists great potential to reduce the load at the source by more aggressively dropping connections. Finding the right balance between sufficient capacity and adequate overprovisioning is the key in this aspect. The system should be able to cope with peer churn and temporal inefficiencies in the delivery process, while keeping the costs for the content provider as low as possible.

The second question asked how delay optimized BTLive is. To answer it, a set of controlled experiments with bandwidth limitations at the source and peers to limit the delivery paths were conducted. It is shown that BTLive is able to maintain both very small startup delays of less than 1.2 seconds for the considered scenarios as well as small streaming delays, as long as the source contributes a minimum bandwidth. In this context it was shown that the recommended minimum source bandwidth of 4 times the video bitrate was not enough to provide a stable system performance, which also can be explained by the proposed theoretical model. For experiments with sufficient source capacities, BTLive was confirmed to have low delays for the studied configurations. Yet, the delay properties require further investigation for larger swarms.

The final question was: What is the overhead of BTLive? To answer this question, the exchanged streaming traffic was studied in more detail. It could be shown that the advertised low delays that can be achieved by the system comes at a high cost, in terms of control and video traffic inside the clubs which is caused by the controlled flooding approach. The announcements used within the clubs to avoid the transfer of duplicate media packets between peers were not able to avoid duplicates. In some cases the number of unique and duplicate video packets delivered to a peer within the club was even the same, i.e. double the data volume was produced inside the club as desired. In future measurement studies it is to be investigated how this problem affects streaming in larger swarms. Especially the envisioned application of BTLive in mobile environments could require a rethinking of the controlled flooding approach inside clubs.

2.5 Analysis of Partial and Social Aware Prefetching in YouTube

YouTube is the most popular streaming platform for user generated content (UGC) in America and Europe according to Cisco [Cis14]. For mobile networks it is the major source of real-time entertainment traffic almost everywhere over the world. They estimate that roughly 6 billion hours of videos are streamed to users each month. About 40% of these are consumed from mobile devices. In the future, as they forecast, mobile traffic will increase nearly 11-fold from 2013 to 2018. This growth is in large parts driven by video streaming traffic. This brings new challenges to mobile network operators, since the data volume is increasing faster than the offered mobile data rates. Even LTE might not fix this problem on the long run [Sol13]. The problem is further amplified by Online Social Networks (OSNs) which offer an easy way to distribute multimedia content, like videos, pictures, and music amongst their users. The concept of subscription to content feeds or channels enables users to get notifications for interesting content items automatically and thereby increases the likelihood that they request these content items within a narrow time frame. E.g., if a video is requested, it is typically played with the smartphone's video player. Due to fluctuations in bandwidth and delay, these players use playback buffering. As shown in [FMM11, MAC+13] there is a potential for optimization as current buffering policies on mobile devices can cause up to 25% of data being transferred unnecessarily, since users tend to abort the playback early.

Recent studies have been conducted on video segment caching and prefetching. An approach that leverages the fact that usually video segments are consumed in a linear manner is presented in [WSL+12]. Cache replacement mechanisms are proposed which are especially beneficial for proxies with small cache sizes. This approach keeps in the cache chunks of videos which most of the users are currently watching or probably requesting soon. The authors of SocialTube [LSW+12] and NetTube [CL09] present a prefix-based prefetching scheme for videos. Their goal is to increase the users' Quality of Experience (QoE), which is sensitive to stalling events at the start of a video [KS12]. YouTube itself provides users with the option to download videos in advance [Anw14]. In the YouTube app available in India, the user can select videos for downloading, which is performed as soon as Wi-Fi is available. Afterwards, the user can play the video from a local cache without using the precious mobile data volume.

The existing works are not able to fully answer the question of which content items, especially videos, and how much of a content item should be selected for prefetching. To this end, this work strives to give an answer to this question by, first, investigating the YouTube video consumption of 700 thousand users from a new dataset to extract relevant usage patterns and, second, draws inferences from the dataset to be used for developing efficient prefetching mechanisms. These mechanisms can be used to identify the right content items to be prefetched for an individual user and to decide on how much (how many segments) and when to issue the prefetching. In this work, real mobile traffic traces, covering a whole country for two weeks are used. To our best knowledge we are the first using real mobile traffic traces, covering a whole country for two weeks. The contribution of this work is two-fold. Predictive features for offloading complete videos on Wi-Fi are proposed and evaluated. As the download of data by Wi-Fi is about 10-times more energy-efficient [GKS+13], our main focus is to perform content offloading to Wi-Fi networks.

2.5.1 Related Work

Due to the use of real mobile traffic traces, covering a whole country for two weeks, this work's results are likely to have minimal biases in the conclusions drawn based on the dataset. Previous research in this area relies mostly on small or fixed- network traces.

A couple of studies have been conducted on partial video caching. Wu et al. [WSL+12] propose, develop and evaluate a proxy caching system which leverages the sequential playback of video segments. The authors consider the client behaviour of all simultaneous users to predict which segments are most likely to be consumed in the near future.

The client behaviour of mobile devices with iOS and Android in combination with the streaming players of YouTube, Netflix, and Hulu is compared by Ahmed Mansy et al. [MAC13]. They investigate the segment length used by the client players with regard to the operating system, the type of connection and the video length. An interesting observation they made is that YouTube relies on progressive streaming for videos shorter than 15 minutes without segment-based delivery. The approach presented in the work mitigates the heavy network load introduced by progressive streaming and strives for lower pausing events during the playback at the same time.

Cheng et al. [CDL08] investigate statistics about YouTube videos, e.g. the length access pattern, their growth trend and the active live span. From the vantage point of a social network they found that YouTube videos build a strong interconnection with small world properties. A video in the related video list is considered as social neighbour of the original video. One remarkable observation is that most of the videos they crawled are marked as music videos by the uploader. Music (22.9%), Entertainment (17.8%), and Comedy (12.1%) are the most frequent video categories observed during their study. 97.9% of the videos had a length of 10 minutes or less. The authors study the active life span of videos, which is the time when videos gain considerable amounts of views. The majority of the observed videos have an active life span of 10 to 100 weeks. In the work presented, a comparison is made between their results based on a fixed network to our large mobile network dataset.

Only a few studies have been conducted on prefetching performed on the user's mobile device. Yet, Khemmarat et al. [KZK+12] propose a recommendation-aware prefetching approach, which was able to achieve an overall hit ratio up to 80% if performed on a network proxy by using 4.76TB memory. Applied on a stationary client device, e.g. a PC, they were able to reach about 50% hit rate. Based on traces from different places, they simulated a video player and the stalling events happening during the playback to motivate prefetching. As their goal is to increase the playback quality, the authors suggest using variable-length prefix prefetching, based on the video properties and the current network environment. Therefore, they define the amount of bytes which should be prefetched as the video's duration times the difference between the video's bit-rate and the download rate. This can be done at the client or by a network proxy. One of their findings is that about 30% of videos were accessed from a related video list and another 30% from the YouTube search result list. Less than 20% of the requests came from YouTube pages and external links, each. Their video selection is based on the search result page of YouTube, where they start to prefetch the top 20 videos of the list. Second, they prefetch the top 25 videos of the related video list when a YouTube video page is accessed. The approach presented in this work also leverages the related videos of a video and goes beyond by considering the partial consumption of videos from different categories.

2.5.2 Datasets

The performed analysis involves three datasets. The first dataset contains real user traces for mobile YouTube video requests. The second dataset contains YouTube meta data for each video in the first dataset. The third dataset contains the first ten related videos for each video in the first dataset.

2.5.2.1 Mobile Network Dataset

This dataset consists of mobile video requests from mobile users for YouTube. It covers the whole area of France and accounts for two consecutive weeks in April 2014. Roughly 10,000,000 requests from more than 700,000 users, accessing more than 1,600,000 different videos are included in the dataset. The data was collected by packet header inspection of TCP traffic in Gateway GPRS Support Nodes (GGSNs[13]), deployed by the network operator. More than 10,000,000 HTTP- GET requests for YouTube videos are captured. Included are requests issued on 2G, 3G, HSPA, HSPA+, and LTE mobile networks. These GGSNs were used by more than 700,000 unique users during the measurement time. For a subset of 3,685,172 requests, the number of requested chunks of the viewers is available. This applies to users requesting videos by Http Live Streaming (HLS[14]) only. HLS is used by devices running Apple iOS if they are connected to 3G networks. Furthermore, for HLS to be used, the video has to be longer than 15 minutes. Additional details about mobile video client behaviour can be found in [MAC13]. For shorter videos, YouTube relies on progressive streaming [MAC13]. These preconditions are met by 465,000 users (65%) and requests (%) for the dataset used.

This information allows analyzing how much of a video has been watched. The collected values are precise to the length of one chunk, since it is not known if the viewer watched the requested chunk completely. This is only a minor imprecision since a chunk is typically 5 seconds long while

[13] http://www.radio-electronics.com/info/cellulartelecomms/gprs/gprs-network-architecture.php

[14] http://tools.ietf.org/html/draft-pantos-http-live-streaming-13

this part of the analysis focuses on videos longer than 15 minutes. Therefore, the implicit imprecision introduced is at maximum 0.6% (5/[60 × 15]) of the video length. It is assumed that the viewer does not jump backwards or forwards during the playback, since the data on this is not available. A further reason for imprecision can be the retransmission of chunks due to bad connectivity and packet corruption. In few cases multiple requests from the same user for the same video within a short time span were observed. These entries are replaced by a single request where the number of requested chunks is summarized and the request time is set to the time of the first request. Reasons for these entries might be that the user has switched the network cell or got connected to another GGSN. Additionally to mobile devices; there are also public hotspots which transfer their traffic through GGSNs. They state a minority of the whole dataset and are removed from the evaluation as they show much higher requests counts than the average users.

2.5.2.2 YouTube Dataset

For each YouTube video ID in the *Mobile Network Dataset*, the available meta data was requested using the YouTube Data API. This meta data consists of the following information: duration of the video, upload date and time, category (as one of YouTube's predefines category chosen by the uploader), rating, number of ratings, number of likes, number of views, number of users which saved the video as favourite, and number of comments. This dataset was collected three months later then the *Mobile Network Dataset* which is expected to lead to different values e.g., for the number of likes and comments compared to the time when the videos were requested by the users. Yet, it allows to better characterize the requests seen in the mobile dataset despite the imprecision caused by this approach.

2.5.2.3 Related Videos

For all videos in the *Mobile Network Dataset*, we retrieved the first ten related videos using the YouTube Data API. The dataset contains about 16 million video IDs together with the ID of the video which they are related to. Previous works showed that these first ten results do not change much, independent from which location they are crawled [KZG13]. The collection of this information happened at the beginning of September 2014 which might introduce videos that did not exist during the trace collection in the related video list for some of the videos.

2.5.3 Trace Analysis

In this section an in-depth analysis of the collected datasets is performed. The emphasis of this chapter is on the YouTube requests on a mobile network. Thereby, a focus is set on observations usable for prefetching mechanisms.

2.5.3.1 Request Distribution

Many users in the dataset have only requested a few videos. About 16.6% of all users in the dataset requested only a single video during the two weeks of data collection. To distinguish the users w.r.t. the number of requests they issued, a CCDF was plotted shown in Figure 45. It is assumed that the user behaviour differs between those who request a high number of videos, compared to those with a low number of requests. Overall, most of the users requested 30 videos or less. These users are considered as *Light Users*. Users which requested more than 30 videos, but less than 100 are considered as *Heavy Users* and state 10.9% of all users. They state a minority of users but are responsible for a large fraction of all video requests. More than 100 requests are issued by 1.8% of the users and considered as outliers. In the dataset, a few users exceeded 1,000 video requests and are most likely Wi-Fi hotspots connected to the cellular network as described in Section

2.5.2.1. Considering all users, the mean request count is 15, the median is 6, and the standard derivation is 27.52.

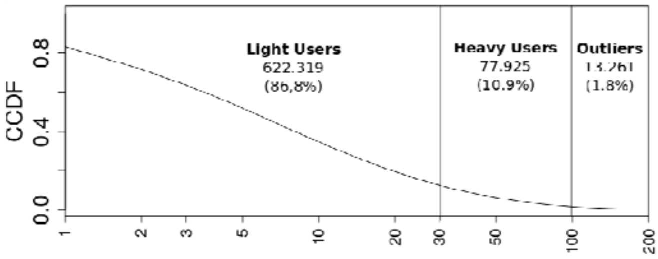


Figure 45: Number of requests

2.5.3.2 *Video Category*

YouTube videos belong to a certain category assigned by the content creator. Most of the videos requested belong to the category *Music* which is not surprising since this was observed before for YouTube video requests in general [CDL08]. However, according to the new mobile dataset, on mobile networks, the portion of music videos is about twice as high and therefore, even more important to consider than in fixed networks. Videos of the category sports are less likely to be watched mobile according to the dataset. This supports the assumption that it is important to consider content meta data when designing a prefetching mechanism.

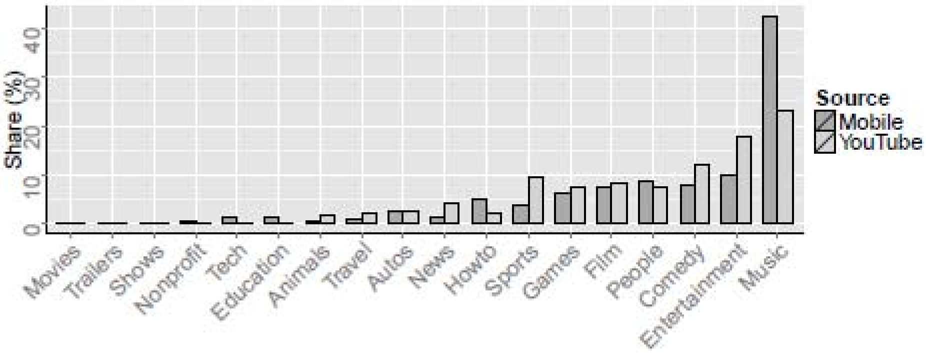


Figure 46: YouTube request distribution compared between all [CDL08] and mobile requests by category

2.5.3.3 *Video Source*

In Figure 47, a CDF of the number of requests made by the users divided by the number of YouTube channels they consumed from is given. There is a clear difference in the behaviour of heavier users compared to light users. While almost 40% of all users watched only a few videos from different channels each, the heavier users tend to watch more videos from the same channel. Figure 47 shows clearly that 50% of the requests from the 100,000 heaviest users had a request/channel ratio greater than 3. This shows clearly that the channels from which videos have been requested previously are useful for predicting the source of future requests. This has been observed especially for heavy users.

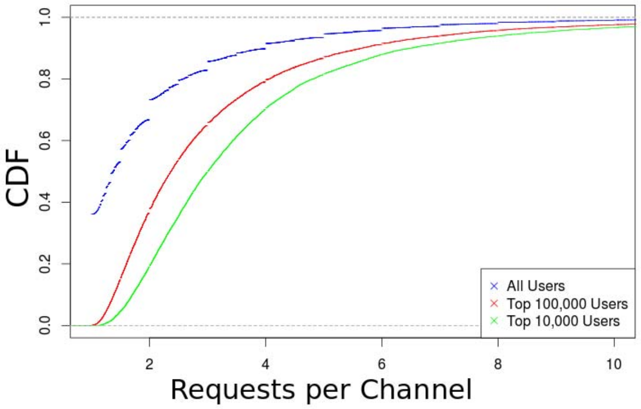


Figure 47: Average ratio between the number of requests and the number of channels for each user classified according to the number of requests.

To clarify this finding, also the single users and their request/channel ratio has been investigated. Many users which requested a lot of videos from only a few channels have been observed. This underlines the potential of the video source, e.g. YouTube channels, as a prediction feature for prefetching mechanisms. Additionally, this leads to the assumption that the user has a special relation to this channel, e.g. has subscribed to it. For those users, it would be beneficial to prefetch videos based on the channel the user is likely subscribed to. Surprisingly, almost every combination of number of requests and number of unique channels consumed from exists. A cloud of combinations observed in the dataset is shown in Figure 48. The blue line in the figure indicates a linear function with an overall minimal distance to all points. It is shown that for, e.g., 100 channels this function maps to 200 video requests. The many points which are on the left side of the figure indicate a huge potential of using the video source as a prefetching criteria.

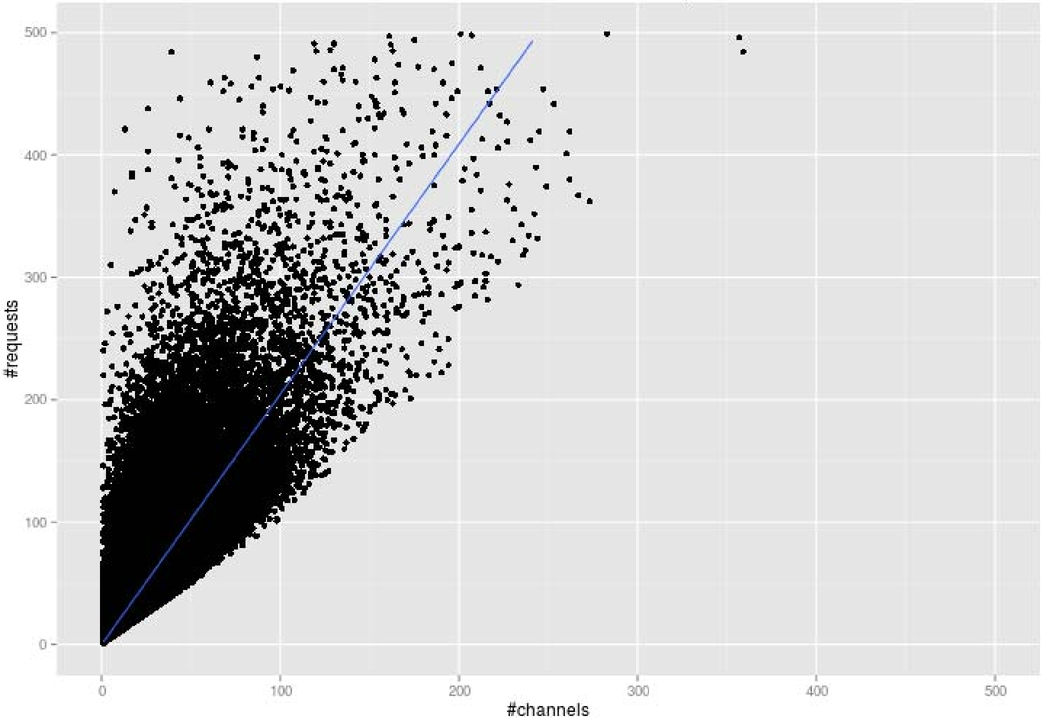


Figure 48: Number of channels vs. the number of requests for each user

2.5.3.4 *Related Videos*

According to [KZG13] about 45% of all videos watched on YouTube are requested from the related videos section on YouTube video website. A dataset of 10 related videos for each video in the dataset was collected to investigate if this observation holds for mobile requests, too. It is likely that the number is lower, because the related video section is placed beyond the video on Android and iOS and to this end not in the user's view while watching a video. Furthermore, the dataset contains only HTTP requests for YouTube videos like, e.g. caused by embedded videos. Videos requested by a browser where the user has logged in his YouTube account or requests issued by the YouTube app are raising HTTPS requests. Therefore, the analysis targets at the consumption of non-logged in users on the YouTube websites or embedded videos.

In Figure 49 an example of a user trace is sketched. Simplified, two types of video requests can be distinguished. On the one hand there are videos from a channel the user is likely subscribed to, indicated by a box containing the letter C. On the other hand, videos which look randomly requested, e.g., because of a custom video search, a link clicked, or random video browsing. In this work the connection between these videos based on YouTube's related video list, which is provided for each YouTube video, has been investigated. Thereby a part of the randomly seeming video requests were identified as videos from the related video section of a previously requested video.

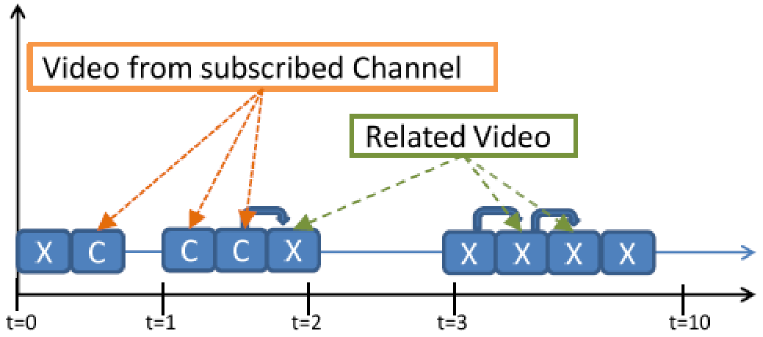


Figure 49: Simplified example of a video trace for a user with explanations: C – video from a channel the user has subscribed, X – random video

Figure 50 shows clear differences between heavy and light users w.r.t. the portion of videos watched based on the related section of the previously watched video. Light users tend to watch very few related videos sequentially, e.g., for 75% of the users 0% of all video requests have been related to the previously watched video. Contrary to the light users, heavy users request much more related videos, e.g. 50% of the heavy users watched about 6% related videos. About 20% of the heavy users watched more than 15% of videos related to each other. This finding shows a potential for predicting content based on the videos a user has requested previously. E.g., assuming a 10% correct prediction ratio, it still is reasonable to prefetch the first 10% of the related videos. This portion can be used as an initial buffer filling if the video is requested and allows downloading the rest of the video while the player consumes from the prefetched portion. Therefore, stalling events are reduced and the users QoE increases.

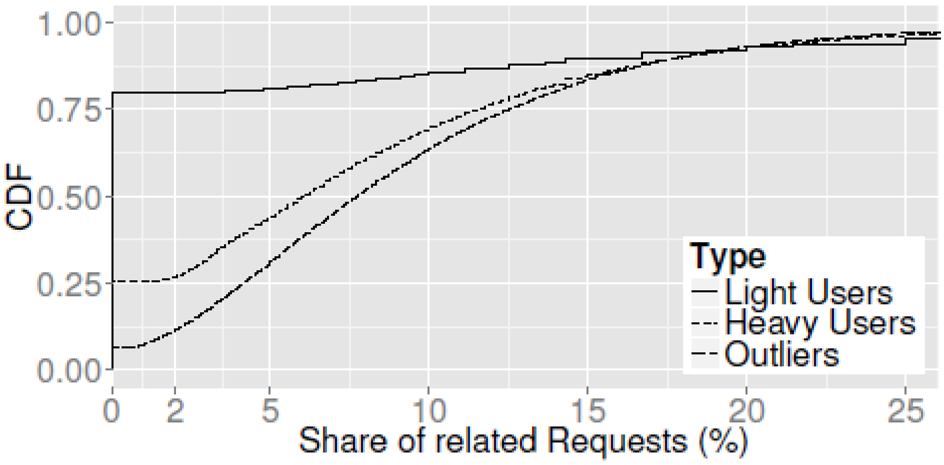


Figure 50: Video requests issued based on YouTube's related video section

2.5.3.5 Content Related Behaviour

Figure 51 shows that most of the requested videos do not have a huge likes or comments count. 50% of the requested videos have less than 1,000 likes. As the effort to write a comment is higher than to like a video, the number of comments is typically less than the number of likes. It has been observed that 50% of the requested videos have less than 100 comments. The CDFs of the comments and likes are parallel to each other after 200 likes and comments. This indicates a stable relation between this two metrics for the videos' popularity. For prefetching mechanism design, this allows to state that videos with fewer likes and comments have a higher probability of being requested. One reason for this might be, that such videos, which tend to have restricted popularity, are often more popular for a close circle of people. Such a social circle can be defined by geographic location, memberships, interests, and content published amongst only a few friends or family.

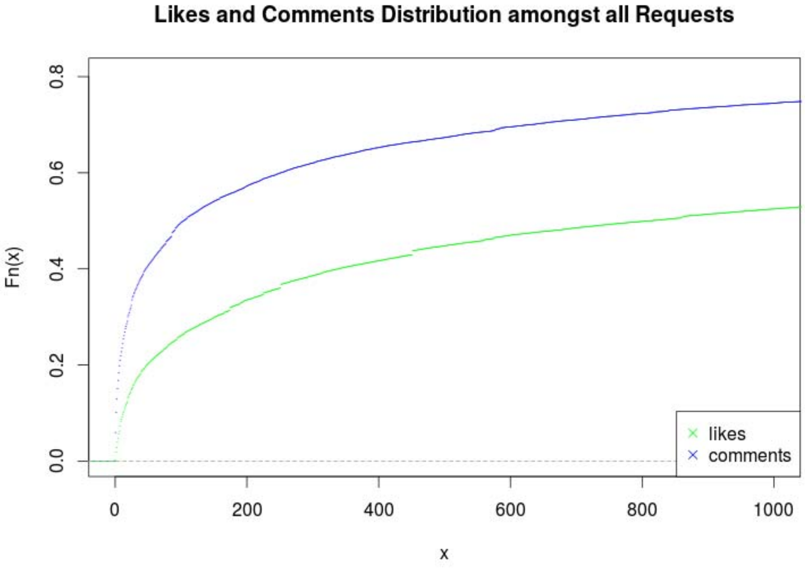


Figure 51: CDF of comments and likes based on all requests

2.5.4 Conclusion

By analyzing a new dataset of mobile YouTube requests, some observations have been made which are useful for the design of a mobile prefetching mechanism. First, it has been shown that videos of a certain category are more likely of being requested mobile. This applies especially for videos of the category music, while videos of other categories, e.g. sport, are more likely to be watched non mobile. Second, especially for heavy users the videos' source has been shown to be a prefetching predictor with high potential, as many videos are requested from the same channels. This feature has to be considered carefully and adapted to a user's behavior since this observation has shown to be predictive for many users but not for all, independent of their number of requests. Third, related videos have been shown to be predictive for a part of the requested videos and might serve as a start point for further research in the direction of partial prefetching since it seems too costly w.r.t. energy to prefetch related videos completely. Fourth, likes and comments turned out to be less predictive for the popularity of the users. The users, e.g. tend to request content with a comment count under 200 in about 60% of all requests seen in the dataset.

2.6 Analysis of Fake Views in Youtube

Online advertising sustains a large fraction of Internet businesses. The International Advertisement Bureau (IAB) [IAB1] recently reported that online advertising generated $42B revenue in 2013, which corresponds to 17% more than in 2012. The dimension of this market brings as no surprise that online advertising attracts fraud. Indeed, fraud is considered to be endemic in the ecosystem: current estimations indicate that 15-30% of ad-impressions are fraudulent [L13, J3, V14] leading to losses in the order of billions of dollars for advertisers [SMS14]. Understanding and containing the impact of this fraud is critically important to maintain the brand value of the online advertising.

Moreover in the area of content distribution, most of the existing solutions such as Content Delivery Network (CDN) based their algorithms on estimating where a content is going to be consumed. Fake views may severely affect these prediction algorithms and lead them to cache/prefetch content in wrong locations, close to where fake views have been produced.

Practitioners and researchers have analyzed and proposed solutions [DGZ13, DGZ12, MAA07, MPG+11, LZX+12, SSZ+11, IAB2, CPA, IMP] to address fraud in traditional forms of online advertising such as search or display ads. With this type of fraud, known as click fraud, the fraudster generates artificial clicks in ads to generate revenues (for instance, in its own websites). Indeed, the typical business model in this case, *Cost-per-Click (CPC)*, monetizes a click in the ad only if it produces a visit to the associated landing webpage (own by the advertiser). Therefore, advertisers have partial information (e.g., the session time in their webpage) that can be used along with other information provided by ad networks to identify fraudulent activities.

As the technology evolves, new forms of online advertising, e.g., mobile ads or video ads, are becoming more popular. Of particular interest to this work is video advertising. Surveys indicate that 93% of marketeers had at some point in the calendar year of 2013 used video for online marketing [eM14], and of these, 65% used specifically YouTube to deliver the video advert. These numbers have strong monetary implications: it is estimated that the online video advertising sector had grown from $2.2B in 2012 to $3B in 2013 and is expected to double to $8B by 2016 [IAB1, em13]. Furthermore, it was estimated to be responsible for approximately 7% of the total revenue generated by online advertisements in 2013 [IAB1].

The profitability of this growing business has attracted the interest of fraudsters in the area of video advertising, which have become very active in the last years resulting in substantial economical losses for advertisers [AW14, K14]. For instance, in December 2012, YouTube removed more than 2B fake views associated to videos uploaded by accounts associated to the music industry [G12, H12]. Moreover, it is very easy to find paid services that offer to generate up to ten thousand views to a video hosted in different popular platforms (YouTube, Dailymotion or Vimeo) at a low price. For instance, in Fiverr.com one can find several of these services.

All these episodes have raised the alarms of the main players of the video advertising industry, including video platforms, whose reputation as advertising venues may be seriously affected, and advertisers, which may incur significant losses if this situation is not properly addressed [SMS14]. Indeed, YouTube, the major player in the online video advertising industry, has explicitly indicated its concern and intention to address this problem [GooS14].

The common attack in video advertising fraud aims at artificially inflating the view count of one (or more) video(s) using bots or other means (e.g., persons in cheap labor countries) to perform fake views on this video. The goal of the attack could be to simply increase the popularity and visibility of the video for marketing purposes or, in case the views are monetized, to generate revenues for the uploader of the video.

The business model used by major video platforms like YouTube or Dailymotion is based on CPV (Cost Per View) so that the advertiser is charged when the user watches the ad. Moreover, the video ad is hosted by the video platform. Therefore, contrary to the case of traditional ads, in this case the advertiser cannot register any information regarding the user activity that could be used to identify fraudsters. Thus, the responsibility (and ability) of identifying fraudulent activity falls entirely in the fraud detection system of online video platforms.

The scenario described above depicts a situation in which we lack a standard way to auditing or monitoring the performance of the fraud detection algorithm used by video platforms. In this work, we first propose a methodology that helps to fill this gap and can be used to evaluate the performance of different platforms. By employing a simple probing technique, our methodology shows that two major online video platforms such as YouTube and Dailymotion seem to present weaknesses to detect fraudulent views.

Based on these initial results, we extend our methodology to analyze key parameters of the fake view detection mechanism of a video portal using a modular automated software. Given the potentially very large space of parameters that can be considered by the adversarial model of video portals, we focus on those parameters that can be directly manipulated by external agents, including: the overall behaviour of an IP address in terms of number of views and number of videos visited per day, the inter-arrival time between consecutive views to a video, the view-duration, the utilization of cookies, etc. In this research we apply our methodology to YouTube since it is, as indicated above, the most important online video platform. Finally, we use two real traces with roughly 20M YouTube sessions collected in two different vantage points of a commercial ISP to validate the penalization (or view-discount) strategy of YouTube from a statistical perspective.

Our main findings can be summarized as follows:

(1) The global behaviour of an IP address (rate of views and number of visited videos) is the main factor the fake view detection algorithm of YouTube uses to discount views. In particular, the most critical parameter is the rate of views coming from an IP address. For a small number of views, and depending on the number of videos across which the views are distributed, YouTube counts all views. However, from a given threshold on, the fake view detection algorithm applies a punishment factor that increases exponentially with the number of performed views. Finally, we observe YouTube applies a more conservative punishment factor, discounting less views, to NATed IP addresses.

(2) Another relevant parameter considered by the fake view detection mechanism of YouTube is the inter-arrival time between views on a video. The algorithm is especially severe with views coming in bursts. Moreover, the detection system uses cookies to identify suspicious behaviours.

(3) The statistics obtained from the real YouTube sessions in our traces indicate that the penalization strategy of YouTube is conservative and seems to be designed to minimize the number of false negatives.

(4) Our methodology reveals that YouTube uses different algorithms to detect fake views when counting the views that are monetized and when counting the views that appear in the public view counter. This seems to indicate that a large fraction of views that YouTube considers suspicious, and thus are discounted from the view counter, are nonetheless monetized.

2.6.1 Background

In this section, we briefly discuss the basic concepts related to YouTube's view counting and monetization systems.

2.6.1.1 YouTube's View Counting System

YouTube associates each uploaded video to a webpage, which embeds a *public view counter* every user can see. Plus, YouTube offers exclusively to the uploaders the *Analytics* service, which provides detailed statistics about their videos.

- Public View Counter: While YouTube updates in real time most of the statistics shown in a video webpage (e.g., number of likes or comments) or a in channel webpage (e.g., number of subscribers), the public view counter follows a different procedure. Specifically, YouTube updates the view counter in real time with every view up to 301 views. Once overcome this threshold, YouTube freezes the counter [Goo14] and starts a validation process of the views to filter out fake views, i.e., views which do not look as generated by a human. Based on our experience, this review process may take hours or even days before letting the view counter overtake the 301 view limit. Once the validation process ends, YouTube updates the view counter with a rate that varies from once every 30 minutes to once every 2 hours [GooD], and it counts only validated views. Indeed, the counter discards all the views, which the validation process labels as fake.
- YouTube Analytics: Alongside the public counter, which only provides the number of views corresponding to a given instant in time, YouTube offers to its uploaders the Analytics service, which collects and shows several statistics with a finer time granularity and over a tunable time window, e.g., the number of views grouped by day, by country, by users' age and gender, by playback location (the user can watch a video directly in its webpage or in another website which embeds it), etc. Besides, Analytics provides the uploader with reports about his channel subscribers, e.g., their likes and dislikes, comments, etc. Analytics updates these statistics once a day [Goo]. Based on our experiments Analytics counts only validated views.

In addition, Analytics provides monetization statistics to the uploaders participating to YouTube's monetization program, i.e., uploaders who agreed to let YouTube attach ads to their videos. These statistics show: (i) the estimated number of monetized views, i.e., the number of users who watched the associated video-ad, (ii) the estimated revenue based on the Cost per Mille (CPM), i.e., the revenue for each thousand monetized views; and (iii) the total gross revenue the video generated. Furthermore, these metrics are available per country, per day and per type of ad.

2.6.1.2 YouTube's Monetization Service

YouTube's data monetization model is very simple. The uploader activates the monetization program for a video, thus explicitly allowing YouTube to associate ads to this video. YouTube supports two different kind of ads: (i) the *Overlay in-video ad*, a banner shown in the lower part of the video, and whose monetization is based on clicks (as traditional banner ads); and (ii) *Video-ads*, which are attached at some point of the video playback (typically before the video starts). In this work we focus on the latter kind of ads, while the former kind is out of the scope of the work. YouTube offers two kinds of video-ads: (i) *TrueView in-stream ads*, which allow the viewer to skip the video-ad after 5 seconds, and (ii) the *Non-skippable in-stream ads*, which the user cannot skip. The video-ads belong to an *advertiser* that uses the Google AdWords service [AW] to launch an advertisement campaign associated to YouTube videos. In the planning of the campaign, the advertiser chooses the daily budget, the maximum Cost per View (CPV), as well as some marketing preferences to target specific audiences, e.g., users' age, country, etc.

When a user starts the playback of a YouTube video, Google runs a complex bidding algorithm to select the ad to attach to the video (which executes in a few milliseconds). YouTube charges the advertiser only if the user watches the video-ad for at least 30 seconds and shares the revenue with the uploader. The charged amount is variable since Google typically follows a variant of a Vickrey auction, named Generalized Second-Price auction [EOS+05] for which the advertiser winning the bid pays the price of the second-highest bid (that varies depending on the video). We remark that the uploaders can access all the statistics about the monetized views associated to their videos in their YouTube Analytics dashboard.

2.6.2 Performance Analysis of the Fake View Detection Mechanism of YouTube

The performance of any detection mechanism is characterized by its rates of false positives and false negatives. In this section, we present a basic methodology to estimate the false positives and negatives ratios of YouTube's fake views detection mechanism and compare them with those obtained for another video portal, which is based on the same business model, namely, Dailymotion.

2.6.2.1 False Positives

In the context of this work, we define false positives as those fake views that are counted as good ones by YouTube. In order to evaluate the rate of false positives, we need to design a methodology that (i) generates (and counts) fake views, and (ii) measures how many of them YouTube labels as good ones. To this end, we design a software based on the Selenium Webdriver [SEL], a library for the testing of webpages that allows to emulate human actions on a browser. Our software loads a video webpage, reproduces the video for a given time and logs the view. In the default configuration of our software, we keep fixed the view duration and the time between two consecutive views.

To estimate the rate of false positives, we use our tool to perform fake views on videos, which we upload to YouTube. Using our own videos allows us to not interfere with the rest of the YouTube system and, thus, make sure that we do not affect other users in the system with our experiments. Moreover, we properly hide our videos (e.g., naming them with random hashes and descriptions and removing external links to them) to prevent other YouTube users to find them and "pollute" our measurements. In this way, we are reasonably confident that all views reaching our videos come from our automatic software. In addition, as uploaders of these videos, we have access to their YouTube Analytics, which provides statistics as the number of views counted by YouTube. Thus, we can compute the false positive rate (Rfp) of YouTube's fake view detection mechanism as:

$$R_{FP} = \frac{\#YouTube\ counted\ views}{\#Tool\ performed\ views} \qquad (1)$$

We pick 70 different PlanetLab nodes [CCR+03], and we divide them in 3 independent sets of different size N = 10, 20 and 40. We assign each set of nodes a different video we uploaded to YouTube, and we configure each PlanetLab node to generate fake views to the video corresponding to its set using our software. We instrument each node to perform 3 fake views per day (one every 8 hours) for a fixed duration of 40 sec each. We conduct these experiments for a total period of 29 days. Finally, we monitor the number of views which YouTube reports in the public counter embedded in the webpage of each of our three videos.

Figure 52 shows the temporal evolution of the aggregate number of views counted by YouTube for each one of the three experiments. As shown, the growth in number of views over time is linear for all of them, and we observe an overall Rfp equal to 97.4%, 100% and 99.1% for N equal to 10, 20 and 40, respectively. From this preliminary analysis we observe that: (i) YouTube's fake view detection system looks not effective in preventing false positives; (ii) it seems that getting fake views counted as real is relatively easy. The latter observation is particularly worrisome, as it implies that an attacker (for instance, with access to a botnet with a large number of IP addresses) could generate a large number of fake views, which could lead to an important benefit for her. For instance, an attacker could "sell" fake views to increase the popularity and the revenue of a YouTube channel, while inflicting important costs on advertisers. Indeed, it is easy to find paid services that offer for a low price to inflate the view-count of YouTube (and other video platforms) videos up to tens of thousand views in a short period of time and at a low price. Some examples of these services can be found at Fiverr.com.

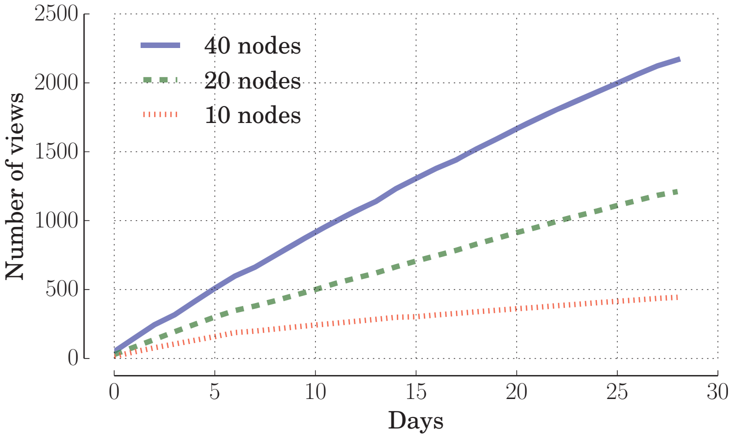


Figure 52: Number of views in the public view counter for three different experiments using 10, 20 and 40 PlanetLab nodes.

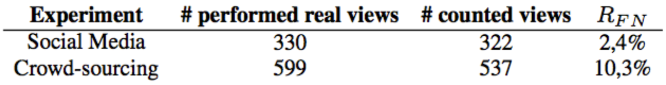
2.6.2.2 False Negatives

In this work we define false negatives as those real views, which YouTube labels as fake, and it discounts from the public counter in the video webpage. In order to evaluate false negatives, we develop a methodology that (i) lets us enroll real users to generate views, and (ii) counts the number of real views to match against the number of views counted by YouTube.

In order to measure the number of real views a video attracts, we embed our video into a webpage we can govern and monitor. Thus, we track the users accessing the webpage, those watching the embedded video and the duration of their eventual views. Based on this methodology, we can compute the rate of false negatives Rfn as:

$$R_{FN} = 1 - \frac{\#YouTube\ counted\ views}{\#Views\ registered\ in\ the\ webpage} \qquad (2)$$

To evaluate the rate of false negatives with the above setup, we conduct two different experiments. For the first experiment, we use social media to announce the URL of our webpage and request collaboration from our contacts and friends. For the second experiment, we use a crowd-sourcing website. The data provided by YouTube Analytics shows in both experiments the typical spatially localized distribution of visits. Indeed, most of the users accessing the webpage are mostly localized in a specific geographical region: in the first experiment most of the views come from Spain, whereas in the second experiment most of the views come from India and Bangladesh. The results of our experiments are shown in Table 15. We observe that he Rfn is reasonably small in both experiments and therefore we conclude that YouTube detection mechanism is able to properly distinguish views coming from real users. Interestingly, we observe that the first experiment presents a smaller Rfn. This difference may be due to the fact that many of the views come from Spain, a country that is reported to present a very low volume of fraudulent activities in online advertising [S10, B13, OFS14].



Experiment	# performed real views	# counted views	R_{FN}
Social Media	330	322	2,4%
Crowd-sourcing	599	537	10,3%

Table 15: False negative ratio for the two conducted experiments.

2.6.2.3 Comparison with other Video Portals

YouTube dominates the marketplace for online video portals. The main competitor following the same advertising-based business model is Dailymotion. Based on some recent statistics YouTube and Dailymotion held 73% and 1.5% of the market share in August 2014 in US [STA14]. Other important competitors are Netflix, Hulu or Vimeo, but they follow different business models based on subscription schemes.

Therefore, for the sake of completeness, we repeat the experiments described in previous sections for videos we upload to Dailymotion. The Rfp also exceeds 90% whereas the Rfn reaches 10.9% and 12.2% for the social media and crowd-sourcing experiments, respectively. Therefore, the performance of the fake view detection mechanism of YouTube and of its immediate competitor are similar. This indicates that the problem of fake views identification is common to (at least) the two major video portals supporting (and funded by) video advertising.

In summary, *there are currently no proper auditing solutions that monitor the performance of the fraud detection mechanism of video portal platforms, and this section has described a first basic methodology to fill this gap. The obtained results reveal that fake view detection mechanisms of the most important video portals perform reasonably well in the identification and counting of real views. In contrast, they show a worrisome under-performance in the detection of fake views*. Performing a complete reverse engineering of these complex detection algorithms is a challenging (if not impossible) task to accomplish. However, even knowing partial elements of them can provide sufficient clues for the design of a second generation of more robust detection mechanisms. With this in mind, in the rest of the work we try to characterize some key aspects of the adversary model and fake view detection mechanism used by YouTube. Note that we focus on YouTube (and leave Dailymotion for future work) due to its major impact on the video advertising industry.

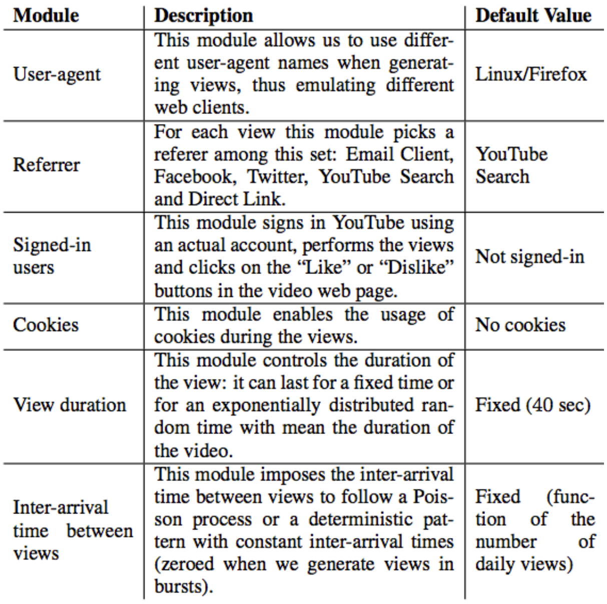
2.6.3 Unveiling Key Elements of the Adversary Model used by YouTube's Fake View Detection Mechanism

In this section we explore different parameters we believe YouTube may employ to detect fake views. This allows us to unveil some fundamental aspects of the behaviour of YouTube's fake view detection system. We acknowledge that the spectrum of parameters and their interactions the detection mechanism actually considers may be too large to make a full exploratory exercise. Instead, we analyze the impact of a subset of meaningful parameters that other detection mechanisms use, and at the same time, an attacker aiming to generate fake views could easily configure to increase the false positive rate.

2.6.3.1 Methodology

To understand the key elements in the adversary model of YouTube, we use a more sophisticated version of the software introduced in Section 2.6.2. This version implements different independent modules, which allow us to configure specific parameters. Thus, by enabling or disabling a module we can study the effect of the corresponding parameter on the performance of YouTube's fake view detection mechanism. Table 16 summarizes the different modules and their main functionality.

To properly characterize the key elements of the fake views detection mechanism of YouTube we conduct an extensive measurement campaign using hundreds of videos during several months. These extensive experiments require the use of hundreds of IP addresses. We collect more than 100 public IP addresses from two different institutions (located in Spain and Germany). Plus, employ 300 PlanetLab nodes. We rely on this large set of IP addresses to deploy proxies to relay the views generated by our software instances running in our local datacenter. Note that we use the transparent proxy software Squid [SQ] that prevents the destination server to identify proxied HTTP requests. Indeed, we observe that YouTube treats equally direct requests and requests proxied by Squid.



Module	Description	Default Value
User-agent	This module allows us to use different user-agent names when generating views, thus emulating different web clients.	Linux/Firefox
Referrer	For each view this module picks a referer among this set: Email Client, Facebook, Twitter, YouTube Search and Direct Link.	YouTube Search
Signed-in users	This module signs in YouTube using an actual account, performs the views and clicks on the "Like" or "Dislike" buttons in the video web page.	Not signed-in
Cookies	This module enables the usage of cookies during the views.	No cookies
View duration	This module controls the duration of the view: it can last for a fixed time or for an exponentially distributed random time with mean the duration of the video.	Fixed (40 sec)
Inter-arrival time between views	This module imposes the inter-arrival time between views to follow a Poisson process or a deterministic pattern with constant inter-arrival times (zeroed when we generate views in bursts).	Fixed (function of the number of daily views)

Table 16: Description of our software modules and their default setting

2.6.3.2 Impact of Different Parameters in the Detection of Fake Views

We leverage the modularity of our software to define different configurations that allow isolating the impact of different parameters on the fake view detection mechanism. In our experiments, each software instance uses a single public IP address chosen from our collection and reproduces a seemingly aggressive behaviour, performing 20 views per day to the same video for 8 consecutive days. The goal of such aggressive behaviour is to guarantee that the detection mechanism of YouTube is triggered. Next we describe the software configurations we use in our experiments:

- Complete (C): This software enables all modules. It generates views using different user-agents, different referrers and some of them are from signed users that sometimes click the *like* button in the video webpage. The duration of the view and the time between views follow a Poisson process. This configuration emulates the closest behaviour to a human being among all the configurations. Thus, we expect its views to be the most difficult to label as fake, adding to the highest false positives rate. This is our benchmark configuration.

- *Poisson (P):* The time between two consecutive views follows a Poisson process whereas we set all the other parameters to their default values.

- *Deterministic (D)*: This setup eliminates any randomness from the instance behaviour by setting the view time and the time between views to constant values, 40 sec and 72 min, respectively. All other parameters take their default values. This fully deterministic behaviour may result suspicious for a detection system and thus we expect a lower false positive rate for its views with respect to the previous configurations.
- *Short views (SV)*: This is a version of the Deterministic setup with a fixed view time equal to 1 sec. We believe that the extremely short duration of the view may raise some further suspicion and thus we expect a lower false positive rate for its views with respect to the Deterministic setup.
- *Burst (B)*: This is a version of the deterministic setup that sets the time between consecutive views to 0, thus generating a burst of 20 consecutive views every day. The time between bursts is configured to 24 hours. Again, views in bursts are typical of misbehaving patterns, and we expect this configuration to show a false positive rate lower than the Deterministic setup.
- *Cookies (CK)*: This last configuration only activates the Cookies module, and let the others take their default values. The usage of the cookies is a standard way for web services to track users' online activity. We rely on this module to understand to what extent YouTube uses cookies in the detection of fake views. We make all the views for each experiment using the same cookie.

For each setup we run an experiment 5 times using proxies located in Spain and Germany. Figure 53 presents the false positive rate for the different setups. The main bar and the error bar represent the average Rfp and the max/min Rfp for each setup, respectively. First, as expected, the *Complete* configuration shows the highest false positive rate (35%). However, surprisingly, it is very closely followed by the *Poisson* (30%). This indicates that the fake view detection algorithm of YouTube neglects the parameters that are not common to these two configurations, i.e., the referrer, the user-agent and the presence of signed-in users. Second, if we compare the configurations with a different setting of the time between views, i.e., *Poisson*, *Deterministic* and *Burst* we conclude that this parameter is indeed considered by the detection mechanism of YouTube. More precisely, there exists a small Rfp difference (~7%) between the cases in which we generate views with random and fixed inter-arrival times. However, YouTube inflicts a severe punishment to bursty behaviours, discounting almost 95% of the views from the *Burst* configuration. Moreover, comparing the *Deterministic* and the *Short Views* configurations we observe that the detection mechanism ignores the view duration despite we expect it to be an obvious candidate for the identification of fake or (at least) not worthy views. Finally, as expected, the detection mechanism leverages cookies, as we observe that the system punishes more severely suspicious, but traceable behaviours.

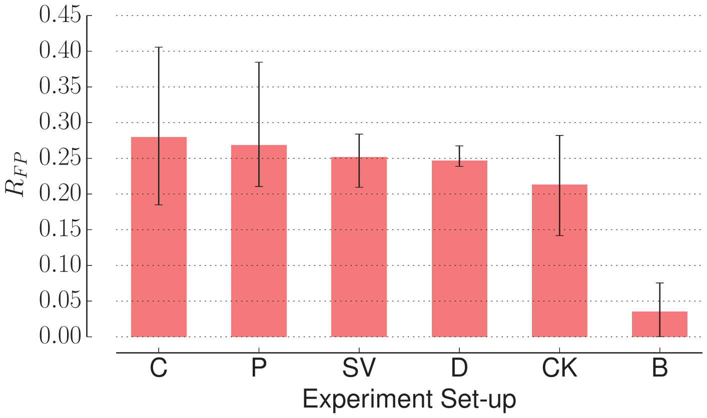


Figure 53: False positive rate obtained for each one of our experiment configurations

In summary, the results in this section unveil that across the studied parameters, the YouTube's fake views detection mechanism leverages solely the time between views and the cookies to identify fake views. While these results explain the Rfp difference between the considered configurations and our benchmark, they do not explain the 65% discounted fake views common to all our setups. This percentage of views is indeed discounted from all our configurations. The only aspect which is common to all our configurations and which may be responsible of such large penalization is that they perform their views from a unique public IP address. This, along with the fact that IP addresses are one of the strongest online users identifiers [CZC14] and one of the key parameters many security online services rely on [MAA07, PSF+07, RF06] lead us to conclude that the video-viewing pattern from an IP address is a key element for the fake view detection mechanism of YouTube. We analyze this hypothesis in the next section.

2.6.4 The Impact of Video-Viewing Patterns on YouTube's Fake View Detection Algorithm

We devote this section to dissect the reaction of the fake view detection algorithm of YouTube to different video-viewing patterns from an IP address using the false positives rate as metric. To perform the experiments described in this section we use the *Deterministic* configuration of our software introduced in Section 2.6.3.

2.6.4.1 Impact of the Number of Daily Views from a Single IP Address to a Single Video

We start our analysis by examining whether YouTube imposes any discounting on the views a single IP address performs to a single video. In particular, we aim at measuring the view per day threshold to overcome before the detection algorithm begins to discount views. Therefore, we conduct a simple experiment in which our software generates W = 1, 4, 7, 8, 9, 10, 20, 30, 40, 50 and 60 daily views to a given video for 8 days. Figure 54 shows the Rfp for the different values of W. We observe that the fake view detection system counts all the views up to a rate of 8 views per day, that seems to be a fixed threshold. From 9 views on, the Rfp decays drastically, as we observe that for 20 and 60 daily views the Rfp is ∼ 30% and almost ∼ 0%, respectively. We can model the Rfp with respect to the daily number of views (W) as an exponential decay function with the following expression that offers an R^2 =0.98:

$$\overline{R_{FP}}(W) = \begin{cases} 1 & \text{if } W \leq 8, \\ e^{-0.1098(n-8)} & otherwise \end{cases} \qquad (3)$$

We repeat the experiment 3 times from IP addresses located in Spain and Germany obtaining similar results.

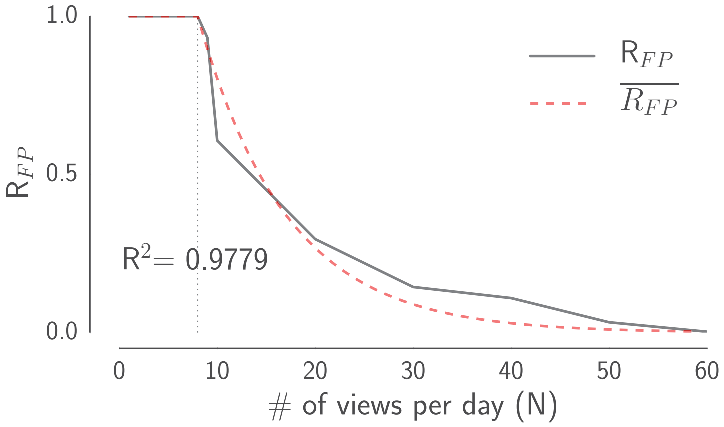


Figure 54: The false positive rate and its fitting model when increasing the number of daily views performed by a single IP address

In order to understand whether the popularity of the video has any impact on the detection of fake views, we also conduct this experiment for two popular videos with roughly 12K (100 in the last month) and 300K (5K in the last month) registered views at the moment we conduct the experiment. To differentiate the views coming from real users and those coming from our software to these popular videos, we configure our software to use rare user-agents (Bada, HitTop, MeeGo and Nintendo 3DS) from which these videos have not received any view in the last 6 months. Indeed, thanks to YouTube Analytics which reports the number of views per user-agent, we can compute the false positive rate associated to views to these popular videos coming from our software. In particular, we run the experiments with W = 8, 9, 10 and 20 daily views finding that also in this case the fake view detection mechanism for these popular videos starts discounting views from 8 views per day. This suggests that the threshold chosen by the fake view detection algorithm of YouTube seems to be the same regardless of the popularity of the video.

It is worthwhile noting that the above observations are in line with the results of the experiments in Section 2.6.2.1 in which each IP address performs only 3 views per day: indeed, this is much below the discounting threshold we observe in Figure 54.

2.6.4.2 *Penalization of Global Behaviour of an IP Address*

In the previous experiment we have analyzed the penalization when an IP address views a single video multiple times. We are now interested in analyzing how the fake view detection system reacts when the views coming from a single IP address are distributed over several videos. We indeed expect YouTube to discount all the views of an aggressive IP address, independently on the number of videos it targets.

In the following we first validate this hypothesis with a simple experiment. Second, we present the results of a large scale measurement study we run to characterize the Rfp as function of the overall behaviour of an IP address defined by the number of performed views and visited videos.

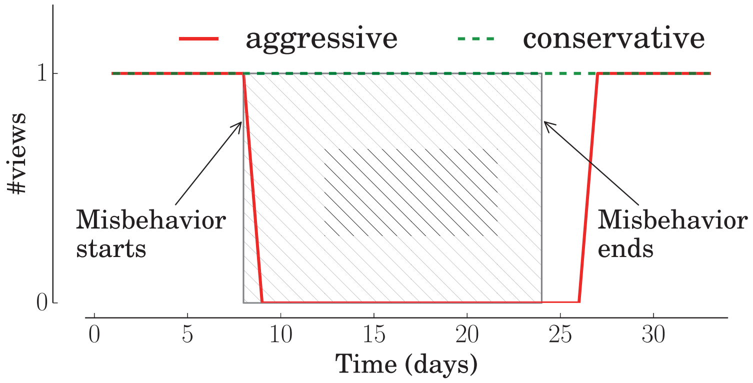


Figure 55: Number of views counted by YouTube for both the conservative and the aggressive IP addresses

- Hypothesis Validation: To validate the above claim, we have performed the following experiment. We run a “good-behaving” instance of our software from a single IP address that performs one view per day for 34 days to a video (Conservative IP address). From a second IP address, we start another “good-behaving” instance, but, additionally, we overlap the watching activity of a second “malicious” instance of our software that performs 25 views per day, starting at day 8 and stopping at day 24 (Aggressive IP address). Figure 55 shows the number of views the fake view detection system counts over time for the good-behaving instances in the two IP addresses. Since our good-behaving instances perform just 1 daily view to the video, this value can be 1 (in case there is not punishment) or 0 (in case there is punishment). The results show that the fake view detection system punishes the aggressive IP address from day 9 to day 26, i.e., the punishment starts the day after the malicious instance starts its activity and it ends two days after we stop the malicious instance. Observe that the conservative IP address does not receive any punishment. Hence, we conclude that YouTube’s fake view detection mechanism punishes IP addresses globally for their overall behaviour, rather than punishing the behaviour for each individual video separately. Note that we repeat this experiment twice obtaining the same results.

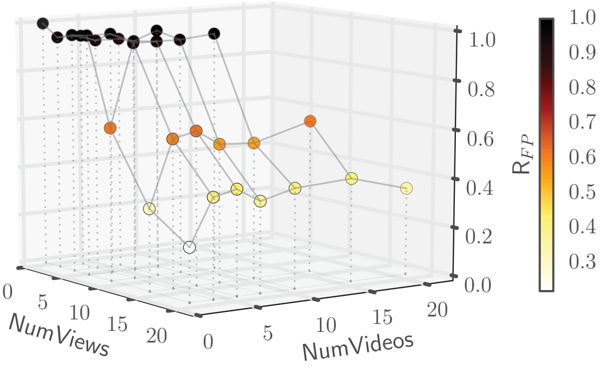


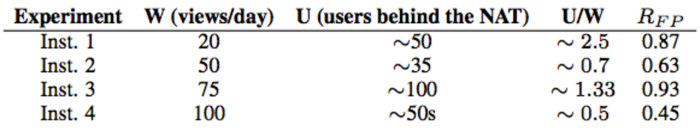
Figure 56: Rfp for several combinations of the number of views W and the number of watched videos D

- Characterization of global IP punishment: The previous experiment shows that when viewing multiple videos, the punishment depends on global behaviour of all the videos. To characterize the punishment inflicted by the fake view detection mechanism of YouTube to an IP address when watching multiple videos, we conduct a large scale experiment in which we perform W = 1, 3, 5, 7, 10, 15, and 20 daily views uniformly distributed across D = 1, 3, 5, 7, 10, 15, and 20 videos during a period of 7 days. In total, we run 28 experiments combining different numbers of views and videos. For each experiment we use a different Planet-Lab node as proxy. Figure 56 shows the false positive rate for each one of the 28 considered combinations. First, if we analyze the evolution of Rfp for a fixed number of videos, we observe the exponential decay reported in Section 2.6.4.1 in every case. However, the threshold in the number of overall daily views that defines the start of the exponential decay varies. It seems to be set to (at least) 10 views for D ≥ 1, whereas we know from Section 2.6.4.1 that it is 8 views for D=1. If we now consider the evolution of Rfp for a fixed number of daily views, we observe that when we concentrate all views in a single video, the punishment is much more severe than when they spread across two or more videos. Moreover, the differences in Rfp for the cases of 2 or more videos are relatively small (≤ 7%).

In summary, *we draw the following conclusions form the results obtained. First, the rate of daily views has a much larger influence on the fake view detection algorithm of YouTube than the number of visited videos. Second, for the number of visited videos YouTube makes a clear distinction between the case in which the views target a single video and the case in which they target multiple videos. Indeed, when we distribute the views across multiple videos we do not observe the same severe punishment we observe when views target a single video.*

2.6.4.3 Are NATed and Regular IP Addresses Treated Equally?

NAT devices aggregate the traffic and, thus, the video viewing activity coming from multiple, usually private, IP addresses into a single public IP address. In large NATed networks such as campus networks, corporate networks and, in some cases, ISP networks, this activity may be significantly large. Therefore, we are interested in understanding how the fake view detection mechanism of YouTube counts the views coming from NATed networks. To this end, we install our software on four machines accessing the Internet from NATed networks located at four different institutions and we configure it to perform 20 (Institution 1), 50 (Institution 2), 75 (Institution 3) and 100 (Institution 4) daily views for a period of 8 days. We remark we use the Deterministic configuration, in which we disable the usage of the cookies. Table 17: Rfp and information about the four scenarios for the experiments we conduct from NATed IP addressesTable 17 shows the Rfp for each experiment along with some information of the different NATed scenarios. Notice that, although our software generates views rather aggressively, the Rfp is surprisingly large in all cases. *This suggests that the fake view detection algorithm of YouTube seems to be able to identify NATed IP addresses and applies a conservative punishment strategy to them*, possibly with the intention of minimizing the false negative rate (i.e., wrongly discounting views from legitimate users). This clearly offers an attacker the chance of generating fake views by gaining access to machines behind large NATed networks, e.g., a public campus network. Finally, we observe a remarkable difference between Rfp across the different experiments that, contrary to our expectation, seems not to be exclusively due to the aggressiveness of the daily view rate. Instead, it seems that the volume of activity originated from the NATed IP address is considered to define the punishment level (note that some other factors that we cannot unveil with our experiments may also be taken into account).



Experiment	W (views/day)	U (users behind the NAT)	U/W	R_{FP}
Inst. 1	20	∼50	∼ 2.5	0.87
Inst. 2	50	∼35	∼ 0.7	0.63
Inst. 3	75	∼100	∼ 1.33	0.93
Inst. 4	100	∼50s	∼ 0.5	0.45

Table 17: Rfp and information about the four scenarios for the experiments we conduct from NATed IP addresses

2.6.4.4 Punishment of IP Prefixes

Our analysis so far shows that an IP address is punished by its global behaviour. In this subsection, we go one step further to analyze whether YouTube's fake view detection algorithm punishes ranges of IP addresses when one of them is misbehaving. Note that punishing IP prefixes due to the misbehaviour of a single IP address is a common technique that, for instance, we have experienced when querying BitTorrent trackers in previous studies [CMC+13]. In addition, some existing solutions propose to consider IP address within the same prefixes as it has been observed that botnet-infected machines choose as potential future members of the botnet machines within the same IP prefix [PFS+07].

We perform a similar experiment to the one described in Section 2.6.4.2. We start an instance from IP address IP-A that behaves properly and makes 1 daily view to a video. After a few days, we start a second instance from IP address IP-B, which misbehaves and performs 20 daily views to a second video. Note that IP-A and IP-B belong to the same /X prefix. We conduct this experiment for values of X ranging between 24 and 30 and we observe punishment only for the case of /30 prefixes. Therefore, we conclude that YouTube detection mechanism punishes consecutive IP addresses as long as one of them misbehaves within a /30 prefix.

2.6.4.5 Detection Time

Figure 57 presents the CDF of the 90th percentiles of daily number of views and of daily number of visited videos per IP address from our traces YT-the 90th percentile of daily visited videos 1 and YT-2. This figure, as well as those of other experiments, indicate that the punishment does not start right after the IP address begins to misbehave. This suggests that YouTube's fake views detection mechanism requires some time before it starts punishing a misbehaving IP address. Our aim in this subsection is to quantify this "detection time" with respect to the past history of an IP address. In particular, we consider three types of IP addresses based on their history: (i) a fully-clean IP address that we have never used to connect to YouTube, (ii) an IP address that we have used before to watch YouTube videos but has never shown a misbehaving watching pattern; and (iii) an IP address that has shown a misbehaving watching pattern in the past. For each one of these IP addresses, we start 7 instances of our software performing W = 3, 5, 7, 10, 15, 20 and 25 views per day, respectively. This aggressive behaviour guarantees that the fake view detection system will mark the IP addresses as suspicious and will discount their views. Our results show that the system punishes the fully-clean IP address after 12 days, whereas it starts punishing the other two IP addresses one day after the experiment starts. Therefore, it seems that YouTube monitors and logs any IP address that connects to the system, and as soon as an already logged IP address misbehaves, the YouTube detection mechanism start discounting its views just after one day. However, for IP addresses which are unknown to the system, the detection mechanism is much more conservative and does not discounts their views until some days have passed.

In summary, the systematic analysis we present in this section reveals some fundamental aspects of YouTube's fake view detection mechanism. We observe that the mechanism uses the IP address as basic identifier and discounts its views *based on its global video-viewing pattern. Furthermore, we show that YouTube punishes two consecutive IP addresses whenever one of them misbehaves. We quantify some thresholds used by YouTube and report the punishment factor for few representative cases. Finally, we provide solid evidences that YouTube detection mechanism uses a more conservative punishment strategy for NATed IP addresses.* In the next section, we use a large-scale trace that includes information of millions of YouTube sessions and discuss (based on the statistical metrics derived from the traces) whether the punishment factors used by YouTube are appropriate or not.

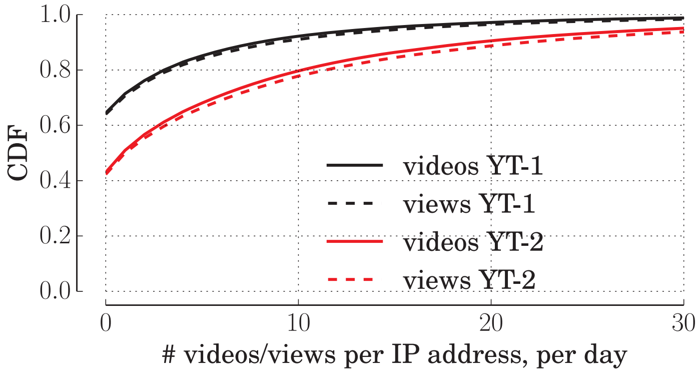


Figure 57: CDF of the 90th percentiles of daily number of views and of daily number of visited videos per IP address from our traces YT-the 90th percentile of daily visited videos 1 and YT-2

2.6.5 Statistical Validation of YouTube's Penalization Strategy

As shown above, YouTube fake view detection mechanism monitors the global video-viewing pattern from an IP address and penalizes those ones overcoming a certain threshold. In the following, we analyze the appropriateness of these thresholds by using a dataset obtained from a large European ISP. Our dataset consists of two TCP-level traffic traces that we obtain by using the traffic monitoring tool Tstat and the trace analysis techniques described in [FMM11].

Trace	Period	Length	IP addresses	Views	Videos
YT-1	01/03/13-30/04/13	2 months	28071	3.94M	1.37M
YT-2	01/05/13-30/11/13	7 months	16781	15.9M	3.95M

Table 18: Measurement traces

Table 18 provides details about the two traces we employ in our study. We collect the two traces at two different vantage points within the same ISP. Measurements have been performed on both incoming and outgoing traffic over two different periods (March-April 2013 and May-November 2013) and together cover approximately 10 months. In total, we observe the activity of about 35K end-users regularly accessing the Internet, and we identify the TCP flows corresponding to almost 20M YouTube video requests and downloads.

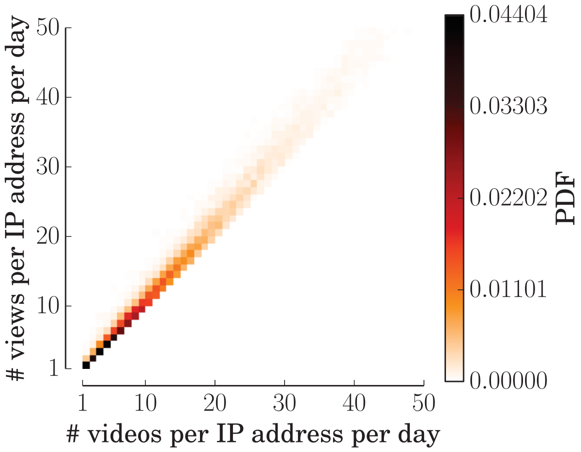


Figure 58: Conditional Probability Distribution of the 90th percentile of daily views given the 90th percentile of daily visited videos from out trace YT-2

We process the collected data to obtain the video-view pattern associated to each IP address in the dataset. First, we notice that the average number of daily views per IP address is 1.29 (3.16) for trace YT-1 (YT-2), whereas the average number of watched videos per day is 1.15 (2.78) for trace YT-1 (YT-2). However, a detection system based on averages would cause a high number of false negatives. As we saw in Section 2.6.2 the YouTube detection mechanism tends to avoid false negatives, thus, for this comparison we decide to use the 90th percentile of the daily made views and visited videos for each IP address as a conservative value to characterize the video-viewing pattern. Figure 57 shows the CDF for the calculated 90th percentiles of daily number of views and visited videos per IP address for both our traces. We can observe that for YT-1 (YT-2) less than 5% (10%) of IP addresses perform more than 10 daily views or watch more than 10 videos. To show the video-viewing pattern of an IP address in more detail, for each IP address in trace YT-2 we have computed the 90th percentile of watched videos per day (D) and the 90th percentile of daily views (W). Figure 58 presents a color map in which the cell (x,y) represents the probability that D equals x and W equals y across the samples. We observe that the region (x,y) ≤ 15 includes all the most likely video-viewing patterns. This confirms that most users make ≤ 15 views per day and watch ≤ 15 videos per day. In addition, we observe a clear linear correlation between the two variables. This indicates that the IP addresses watch between 1 and 15 videos per day and watch each of these videos only once in 91% of the cases. We observe similar results in trace YT-1.

This statistical analysis gives us a good indication of the "typical" video-viewing pattern of YouTube users and thus provides useful information to define an appropriate penalization strategy. For instance, it is very unlikely that an IP address makes more than 20 views or watches 20 videos per day; therefore, if we repeatedly observe this video-viewing pattern in an IP address, a high view-discount factor should be applied.

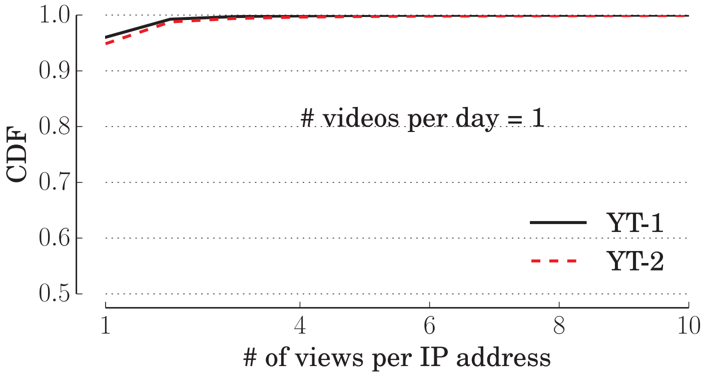


Figure 59: The CDF of the number of views performed by the sample of IP addresses viewing exactly one video in any given day for our traces YT-1 and YT-2

Based on the above initial observations, it seems that the view-discount factors unveiled in Section 2.6.4 for few representative cases are reasonable, as they severely penalize watching-video patterns that perform a large number of views. To further validate this claim, we analyze the appropriateness of the discount factor for the number of daily views to a single video (shown in Figure 54). To this end, we have extracted all the sessions in our trace YT-2 that watch a single video (once or multiple times) in a day, and show the distribution of the number of views in Figure 59. This result shows that for both our traces more than 99.9% of the IP addresses watching a single video in a day perform ≤ 8 views to that video. This confirms that the view-discount factor used by YouTube for the number of daily views to a single video seems to be appropriate to penalize outlying behaviours, thus minimizing the number of false negatives.

From the above results, we conclude that YouTube seems to follow a generally appropriate view penalization strategy to identify outliers and minimize the number of false negatives. This seems to be aligned with the results from the performance evaluation we present in Section 2.6.2.

2.6.6 Detection of Fake Views for Monetized Services

Our analysis so far has focused on understanding how YouTube interprets and discounts the views displayed by the public counter. Since inflating the view counter is amongst the most obvious of threats to YouTube's Monetization Program, in this section, we look at how fake views are handled in videos enrolled in the monetization program.

To do so, we enroll 7 different videos in YouTube's Monetization Program. Of these, we enroll four videos (1, 5, 6 and 7) on to the program from the start, and we enroll the remaining videos (2, 3 and 4) after they attain 301 views. To reduce the probability of our videos attracting legitimate user attention, as indicated in Section 2.6.2, we use random hashes for their names and descriptions and make sure that they are not externally linked. Then, we instrument a given instance of our software (on a unique IP address) to access each video with a daily view rate equal to 20 views per day for the duration of the experiment (64, 27, 33, 33, 20, 20 and 20 days, respectively). Finally, when accessing the video, the instance watches the complete video (at most 194 sec) as well as any ads presented with it entirely. We remark that we have not received any money while running these experiments and all the statistics we report are those we retrieve from the YouTube Analytics channel page.

Figure 60 presents, for each video, the total number of views our tool performs (W) along with the false positive rate we compute from both the YouTube Analytics counter and the number of monetized views (also reported by YouTube Analytics). Surprisingly, our results indicate that the number of views YouTube Analytics counts is significantly smaller than the number of monetized views for all videos. Indeed, the false positive rate ranges between 16.9-33% for the number of views counted by YouTube Analytics and 87.2-94.9% for the monetized views, respectively. Since YouTube Analytics counters are those presented in YouTube's public view count for videos, *this suggests that the algorithm used to detect fake views is much more conservative for the public view counter than for the monetized views. In other words, a large fraction of views that YouTube labels as suspicious, and are not counted by YouTube Analytics for the public counter, are still monetized*. Additionally, we do not observe significant differences between videos that have not reached the 301 views threshold and those which have overcome it.

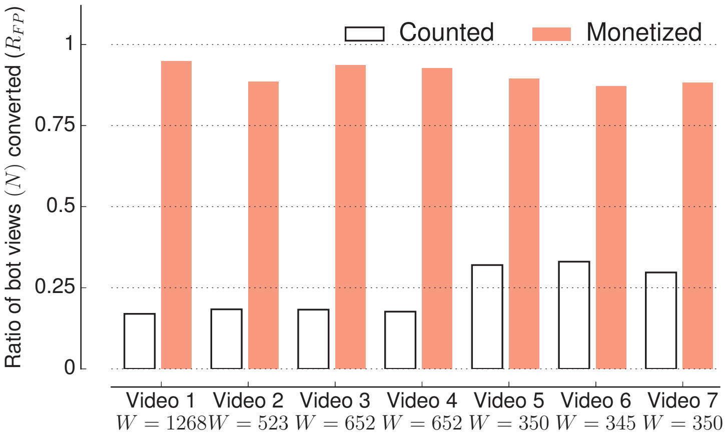


Figure 60: Comparison of false positive ratio for the number of monetized views and the number of counted views by YouTube for 7 different videos

This unexpected result has also been reported by some YouTube video uploaders in the YouTube Analytics page. They report that during some periods of time, the number of monetized views is larger than the counted views. The answer from YouTube suggests that when the number of monetized views is higher than the number of counted views, this may be due to users watching the ad but not the video, and as a result such views would be monetized but not counted for the public counter. However, this claim does not hold in our case, since (as mentioned before) our software instance watches both the video-ad and the video entirely.

Finally, we highlight that we have reported the above findings to YouTube and we plan to present their feedback and explanations, once we receive them, in the next version of this work.

2.6.7 Related Work

The research community has devoted an important amount of effort to the identification of malicious behaviours in online services and to the design of counter-measures to such behaviours [SAM12, CJB08, VBS]. Similarly to YouTube's fake view detection mechanism, most of the detection system designs rely on the IP address as the main id (identifier) to track and identify malicious behaviour. Some examples of such mechanisms are the classical monitoring tools looking for sources of attacks, such as port scanning [SHM02] and DDoS attacks [PLR04], or the detection systems which counteract malicious users in P2P applications [CKG+14]. Only those systems requiring the user registration to gain access to the service, e.g., Online Social Networks, implement detection mechanisms that use both the IP address and the user-id as basic units to detect inappropriate behaviours. For instance, Facebook traces the requests pattern from a given account and if it is unusual the user is warned and if the behaviour persists the account is closed [GKB11]. In the case of YouTube, both registered and non-registered users can access to the service, although as our results suggest it seems that the detection algorithm does not make distinction between both types of users unless cookies are enabled.

More recently, the rapid proliferation of botnets and specialized bots to attack specific services has led the research community to work on the identification, characterization and elimination of botnets and bots [KRH07, XYA+08, LCW10, SKV10, ZP11, ZZY13, YHG11]. Additionally, following a similar methodology to the one we use in this work, Boshmaf et al. [BMB11] and Bilge et al. [BSB09] create their own automatic software to evaluate the efficiency of the detection and defense of different social networks from different types of attacks such as user impersonation.

In the field of fraud detection and mitigation in online advertising, most of the literature focuses on traditional type of ads such as search or display ads. In this case, the fraud problem is referred to as click fraud since the fraudulent activity is associated to fake clicks on ads, typically performed from bots. Metwally et al. [MAA07] present an early study in which they use the IP address as the parameter to detect coalition of fraudulent users or *fraudsters*. In a more recent work, Li et al. [LZX+12] propose to analyze the paths of ad's redirects and the nodes found in the content delivery path to identify malicious advertisement activities. Furthermore, Stone-Gross et al. [SSZ+11] managed to get access to a command-and-control botnet used for advertisement fraud in which the bot master sends commands with fake referrers. On a complementary work, Miller et al. [MPG+11] study the behaviour of two clicking robots: Fiesta and 7cy. Fiesta uses a middleman that probably shares its revenue with advertiser sub-syndicates. 7cy tries to emulate a human behaviour and presents different behaviours depending on the location of the infected computer. Finally, Dave et al. [DGZ12] design an algorithm to identify click fraud from the advertiser perspective; to design this algorithm, the authors propose to measure different aspects of the user behaviour in the advertiser webpage such as the mouse movements or the time spent in the website. Based on their initial work, the same authors propose, implement and test ViceROI [DGZ13], a solution to discount fake clicks from ad networks. The basis of ViceROI detection algorithm is the fact that click-spammers will lead to a higher ROI (Return of Investment) than a legitimate publisher, as the authors claim that a realistic ROI is difficult to obtain with robots.

All the above works establish a very solid basis for the design of tools to mitigate fraud associated to traditional ads. However, they are (in general) not applicable to fraud associated to video-ads due to the different nature of video-ads and click-based ads. To the best of the authors' knowledge, there is only a very recent study that analyzes fraud in video-ads [CZC14]. The authors of this study use traces from a video platform in China to identify statistically outlying video-viewing patterns and, based on these observations, suggest a fake view detection algorithm built on parameters such as the number of views made from an IP address to a video or the number of different IP addresses watching a given video. Unfortunately, as the authors acknowledge, they do not count with a ground truth dataset to validate their designed solution. In contrast to this work, our study focuses on YouTube, the most important video platform worldwide, and pursues a different goal. We propose a methodology to generate ground truth scenarios so that we can evaluate the performance (and unveil basic functionality principles) of YouTube's fake view detection mechanism for both the number of counted and monetized views. As our methodology is extensible to other video platforms, the authors from [CZC14] could use it to validate their proposed solution in their considered video platform. Finally, although less related to our work, it also is worth referring the reader to a recent work by Krishnan and Sitaraman [KS13], which presents a large scale analysis of the different factors that influence the effectiveness of video-ads.

2.6.8 Conclusions and Future Work

The increasing fraudulent activity in the area of online video advertising has raised concerns among the affected players, including advertisers and video platforms. The advertising business model of online video portals limits the means to detect fraudulent activity (i.e., fake views) to the detection systems of video portals. However, there are not standard auditing or monitoring techniques to measure the performance of these detection systems. In this work we have presented a methodology based on active measurements that may serve as an initial tool to advertisers, regulators, etc., to assess the performance as well as to understand some key elements of the fake view detection system of a video portal. We applied this methodology to YouTube that is the most important online video portal in both number of visits and advertising revenue.

Our results indicate that the detection system of YouTube monitors the behaviour of an IP address in terms of the rate of views and the number of visited videos. If the rate of views is small (in the order of 8-10 daily views) the detection system does not discount views. However, higher rates of views trigger a punishment (or views discount) factor that grows exponentially as we increase the rate. Interestingly, this punishment factor is lower when views come from a NATed IP address. Moreover, the detection mechanism also takes into account the inter-arrival time between views, applying a severe punishment factor for bursts of views on a video. Finally, the detection mechanism takes advantage of cookies to trace users, and applies larger discount rates to the suspicious ones. The validation of the punishment strategy of YouTube conducted using the statistical information from 20M YouTube sessions indicates that YouTube's strategy is globally reasonable and seems to aim at minimizing the number of false negatives.

Finally, our experiments provide solid evidences suggesting that YouTube is monetizing views that, on the other hand, it discounts from the public view counter. This indicates that the punishment strategy used to discount monetized views is significantly softer than for discounting views from the public counter.

As immediate future work we plan to apply this methodology to other online video portals starting by Dailymotion. Moreover, we plan to periodically apply our methodology to YouTube and other video portals in order to monitor the evolution of their detection systems. In parallel, we will work in the design of more robust detection systems that would implement obvious functionalities missed in current systems (e.g., discount views that watch a video less than X sec, X = 1 or 2). Moreover, we will explore the utilization of machine learning theory for the identification of outlying video-viewing patterns, not responding to human behaviour as well as the application of signatures that may serve to identify common malicious behaviours across multiple IP addresses.

2.7 Exploring D2D Opportunity using Orange Traces

Device-to-Device (D2D) content delivery can only be effective under certain circumstances. On the one hand, a certain *content popularity* is necessary; on the other hand the *device density* has to be high (like in urban areas). These two factors are essential, so that it is likely that at least one device in range is able to serve the desired content, even when using an optimal device-to-device discovery and delivery strategy.

For further research on D2D content delivery, it is of interest if the quota of requests that may be served via D2D is high enough to justify the implementation of D2D mechanisms in mobile devices. In this project, we suggest an evaluation mechanism to give an answer to this question, based on real-world measurement traces of mobile devices.

The **benefit** of different stakeholders from participating in a D2D content delivery system is discussed in Deliverable 2.3 (Section 3.3). The quota of requests that can be served via D2D is the basis to quantify the benefit of users and their overall, direct **incentive** to participate in a D2D scenario under certain conditions: with growing quota of D2D-retrievable content, the infrastructure expenses of the user are reduced accordingly. Thus, if the quota is sufficiently high, this reduction justifies the implementation and overhead expenses induced by D2D.

2.7.1 Input Data

The input data can be obtained from tracing users of a large mobile network operator. The evaluation is based on two datasets (e.g. CSV files), the **position** and the **request** samples. Both datasets have to be collected over the same time period.

The **position** samples are based on a user's cell associations. We do not use exact positioning (e.g. GPS), as it is too privacy-sensitive, hard to anonymize, and hard to obtain. At random but frequent time intervals (not more than 15 minutes apart if a user is online), the dataset contains information for every user being associated to a specific cell. It is comprised of the following fields:

- The current **time**. It must be accurate, and it can be relative (e.g. 0 = start time of experiment).
- A **user ID**. The ID is anonymized by using any random unique number for a user.
- A **cell ID**. The ID uniquely identifies a mobile base station. The cell ID can also be anonymized like the user ID, without any geographical information.

The **request** samples contain information about a particular user consuming content. It holds the information whenever a user requests content.

- The current **time** of content request, which can be relative (e.g. 0 = start time of experiment).
- A **user ID**. The ID is anonymized by using any random unique number for a user, but correlated to the request dataset.
- A **content ID**. The ID uniquely identifies content. It can also be anonymized by using a random unique number, without any further information about the content.

We can evaluate the quota of successful D2D deliveries like if the users in the trace data had used D2D content delivery in reality, by *simulating* it with the following model.

2.7.2 Delivery Model

We use a **one-hop** delivery model. This means, a device can retrieve content only from devices in direct communication range that have this content stored for own purposes. There are no "selfless" or artificially rewarded intermediate devices that relay content for others.

Our model assumes a **prediction** mechanism which can tell if the user is likely to consume a certain content item (video/audio file, application) in the future. In our model (Figure 61), a prediction can be made at most $\mathbf{t_p}$ before the time of the content consumption $\mathbf{t_{req}}$. This time interval is the *prediction phase*. After $\mathbf{t_{req}}$, the content remains in the device's **cache** for $\mathbf{t_c}$ until it becomes unavailable (the caching phase).

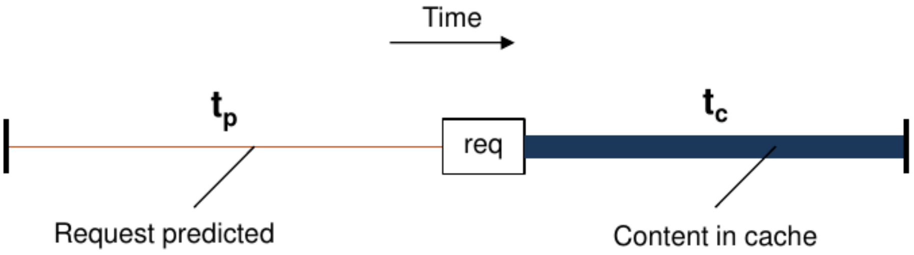


Figure 61: Assumed model for device-to-device content delivery

In our model, a *request* of a device for content is the point in time where the user wants to actually consume the content, i.e. when it is needed. We assume a device **d** makes a request $\mathbf{req_d}$ for a content item at time $\mathbf{t_{req\_d}}$. If at any time $\mathbf{t_{enc}}$ in the *prediction* phase of $\mathbf{req_d}$, a device **e** was in communication range (proximity) of **d** which requested the same content and is currently in the *caching* phase of the request for this content, then (Figure 62):

- A D2D content delivery of a device **d** for a request $\mathbf{req_d}$, is assumed to be successful.
- The start of the caching phase for $\mathbf{req_d}$ is shifted back in time; it does not start at t_{req_d} but at $\mathbf{t_{enc}}$. This is because in a successful D2D transfer, the content is assumed to be transferred at the first time the devices are in proximity, and remain in the cache until requested (and longer).

Otherwise, the D2D transfer is assumed to be not successful (as there is no device in range having the content cached). The caching phase then starts at the time of request, as in this case the content is assumed to be retrieved on-demand via infrastructure. In either case, the content is stored for $\mathbf{t_c}$ after the request.

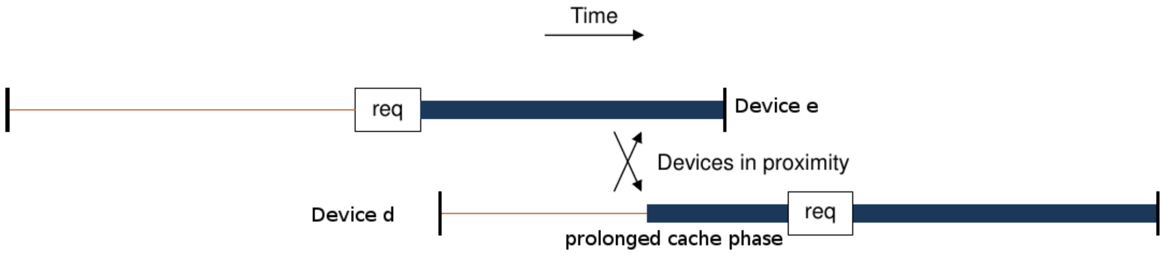


Figure 62: Example of a successful D2D content delivery

2.7.3 Proximity Model

It is assumed that peers (devices) are in communication range, if they are in the same mobile **cell**. This is only an approximation to real-world scenarios, as two users in the same cell may be in range of the cell, but not in range to each other.

No continuous cell join/leave data is supposed, but instead time samples. Peers are in range to each other if the following two conditions hold:

- Both peers generated a sample in the cell.
- These samples are at most $\mathbf{t_{st}}$ (Sample Tolerance) apart from each other. This is a global time parameter, by default set to 10 minutes.

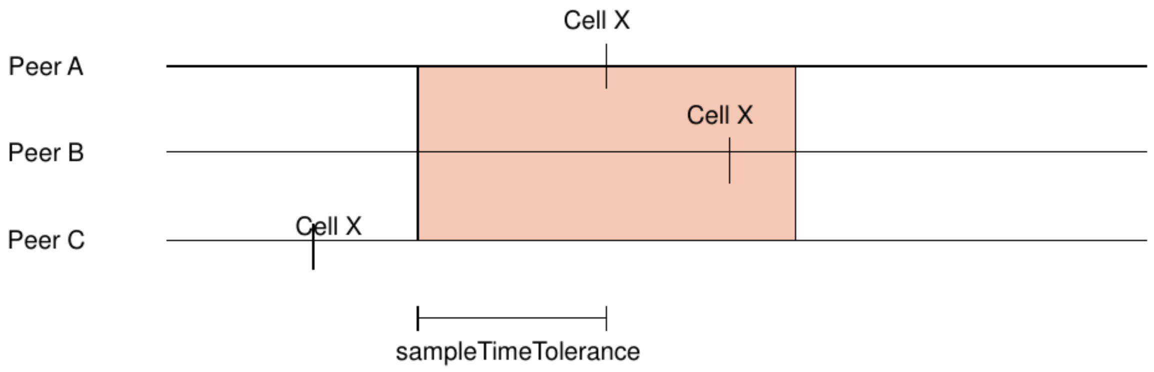


Figure 63: Example of the proximity model: peer A and peer B are in communication range, peer C to neither peer A nor peer B.

2.7.4 Evaluation Algorithm

First of all, the following algorithm is executed **separately for every distinct content item**. It works on a queue **Q** of request objects (containing all requests of all devices for the particular content item). A request object **r=($\mathbf{t_r}$, $\mathbf{t_b}$, d, p)** consists of a parameter $\mathbf{t_r}$ which is the time of the request, and a variable parameter $\mathbf{t_b}$, which resembles the time the content item starts to be in a device's cache. Initially, $\mathbf{t_b}$**=**$\mathbf{t_r}$. The queue **Q** is **always ordered** by the variable $\mathbf{t_b}$ in a device's cache, this means it may be reordered if it changes. **d** is a Boolean parameter which expresses if this request could be served via D2D. It is initially **false, p** is the peer that does the request. The function **prox($\mathbf{t_1}$, $\mathbf{t_2}$):** returns a set of tuples **(p, t)** with all peers in the same cell between time $\mathbf{t_1}$**,** and $\mathbf{t_2}$, according to our proximity model, and the corresponding time of proximity.

The following pseudocode algorithm describes our method of evaluation:

```
d2d_requests = 0
total_requests = 0

While (Q not empty):
        Get first element r=(tr, tb, d, p) from Q
        For all peers (p_prox, t_prox) in prox(tb, tr + tc):
                Is there an element r2=(t2,tb2,d2,p2) in Q where, p2==p_prox && t_prox < t2 < t_prox + tp?
                If yes:
                        d2=true (Mark content as D2D-served)
                        tb2 = t_prox (set tb to time of proximity)
                        Break the for-loop.
        If d=true: d2d_requests++
        total_requests++
quota = d2d_requests / total_requests
```

The result is the quota of possible requests that can be served via D2D.

2.7.5 Evaluation Implementation

We implemented the evaluation algorithm on the Java Platform and Language. We used Java Collections to reduce complexity in map, list and queue operations. Our simulation server is based on Debian with 128GB of memory. A run on the dataset (approximately 80 million entries) currently lasts about 30-40 minutes.

In future work, we can classify the cells (urban, rural). Furthermore, the proximity model may be refined.

2.8 Analysis of Datasets: Orange's CUTV, TSP's FB Data, Flixter

In this section we investigate which model is best suited to capture the correlations in various datasets. The knowledge which model fits the dataset best gives us insight how to design better pre-fetching and caching algorithms.

We consider two kinds of datasets: datasets wherein users gave a rating to content items and datasets expressing which content items users actually consumed. In some datasets there is also timing information, but we ignore this timing information in this section. Instead we randomly split the dataset in a training set and a test set (of roughly equal size). The training set is used to tune the parameters of the model, while the test set is used to assess the performance (which will be detailed below). The model that has the best performance on the test set (being trained on the training set) is considered to be the best model.

The models that we consider are extensions of the models introduced in Deliverable D4.2 [D4.2].

The first model that we consider assumes that each user n and each content item k can be characterised by an L-dimensional vector, where L is a parameter. The rating that user n would give to item k or the likelihood that user n will consumes content k is given by the dot product of the vector associated with user n and content item k. As in Deliverable 4.2 [D4.2], the vectors associated with users are the rows of a matrix P, while the vectors associated with the content items are the rows of a matrix Q, so that the (n,k)-th entry of the matrix product P·Q' is the prediction of how much user n likes item k. The matrices P and Q are determined on the test set. In deliverable D4.2 [D4.2] we also introduced a regularisation parameter (that forces the entries of P and Q to be small), but we found there that it has not a lot of impact, so that here we omit it.

The second model that we consider assumes that the reaction of a user n towards content item k can be predicted as an average over the reactions of a subset of users and content items. Which subset of the users and content items is used is determined by the correlation that exists in the training set between users and content items respectively. The community of user n is captured in the n-th row of matrix C. The (n,m)-th component of this matrix C is 1 only if user m is in the community of user n and is 0 otherwise. Likewise the genre of item k is captured in the k-th row of matrix G. The (k,l)-the entry of this matrix G is 1 if item m is sufficiently similar to item k and 0 otherwise. How user n will react to content item k, is proportional to the (n.k)-the element of the product C·R·G'. Notice that this is an extension with respect to deliverable D4.2 [D4.2], where G was the identity matrix. We construct C via determining the N_S users that have the highest correlation with user n and G via determining the K_S content items that have the highest correlation with content item k in the training set (where N_S and K_S are the parameters of the model).

When we predict user ratings we use the RMSE (root mean squared error) as performance metric. When we predict whether or not a user will consume a content item, we compare the predicted likelihood of consumption with various thresholds. For each value of the threshold, we determine the CP (correct predictions) and UP (useless predictions). The model that provides the best trade-off between those two metrics is the best model.

2.8.1 Orange CUTV Dataset

The Orange CUTV dataset (see deliverable D3.2 [D3.2]) captures which user n watched content item k at which time. Its overall characteristics are described in deliverable D4.2 [D4.2]. As discussed we ignore the timings, to determine the structure in this dataset. In Figure 64 the performance (in terms of the UP vs CP curve) of both models is shown. For both models the optimal parameters were determined. For the first model it was L=20 and for the second N_s=K_S=40. It can be seen from this figure that the first model outperforms the second.

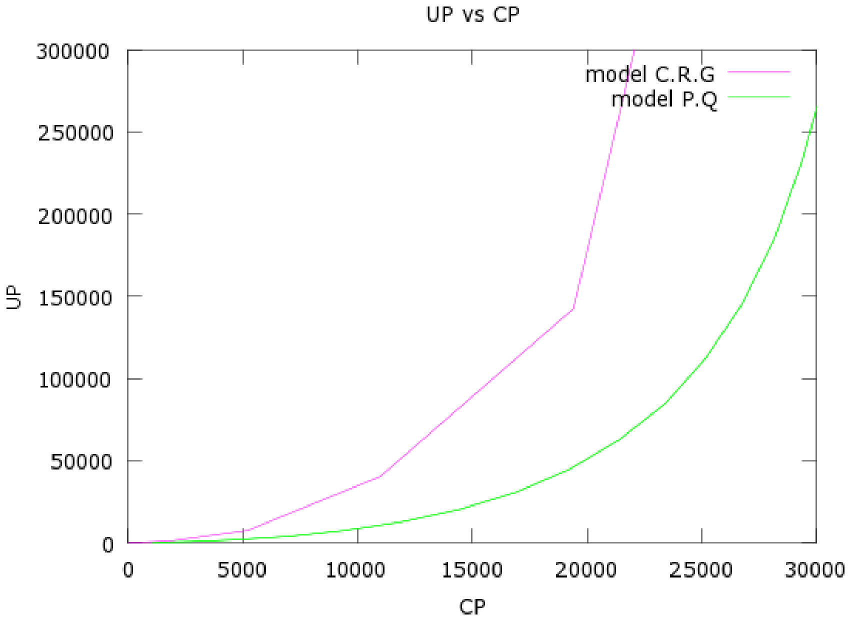


Figure 64: Comparison of both models on the CUTV dataset.

2.8.2 Flixter Dataset

The Flixter dataset [FLIX] contains the rating users gave to content items. Its broad characteristics were described in deliverable D4.2 [D4.2]. Figure 65 illustrates the impact of the number of features L on the RMSE of the training and test dataset. Notice that as the number of features increases, the RMSE on the training set keeps decreasing, but that while there is an initial decrease of the RMSE on the test set, it starts increasing if L gets too large. In that region of large L there is over-fitting. The figure shows that the best value for the number of features L is 5, and that the minimum is very broad. It also shows that choosing the value of L a little bit too large is less damaging than choosing the value of L too small.

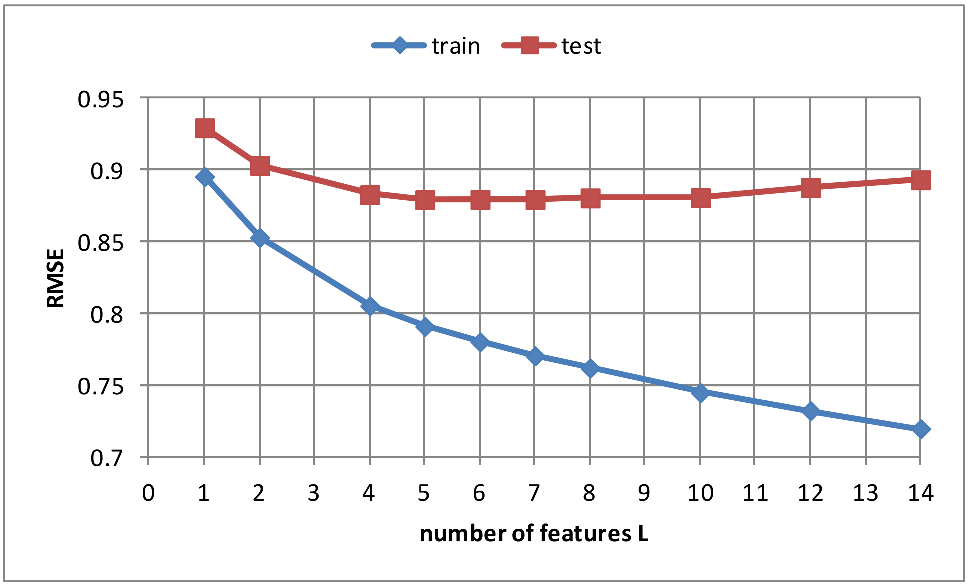


Figure 65: Best parameter for the first model for the Flixter dataset.

2.8.3 Conclusion

In this section we proposed two models to predict consumption or appraisals and analysed their performance on some dataset. The model based on learning a feature vector for each user and each content item and where the prediction consists of a dot product of the feature vector associated with the user and content item, outperforms the model in which the prediction is based on making an average within a set of close users and content items.

3. SOCIAL-AWARE CONTENT DIFFUSION

Based on Tasks 3.1 and 3.2 as reported in [D3.1] as well as Sections 1 and 2 above respectively, the goal of Task 3.3 of WP3 was to predict aspects related to content diffusion, e.g., prediction of content popularity, content consumption patterns (where a content is expected to be consumed), and social cascades of content taking into account the social structure of users and their personality, e.g., estimated through language analysis. Additionally, Task 3.3 analysed the impact that the social dimension has on content interests within the users social graphs to exploit potential content diffusion.

An initial set of mechanisms for prediction of content popularity and consumption patterns as well as the social impact on content interests was documented in [D3.1]. Accordingly, this section provides the final update on social-aware content diffusion approaches. The outcome of this work includes a microscopic information propagation analysis in Google+ and its comparison with Twitter, as well as macroscopic geographical information propagation analysis in Twitter.

3.1 Microscopic Information Propagation Analysis in Google+ and its Comparison with Twitter

Information propagation is an inherent property of human beings that are continuously retransmitting and sharing the information they receive with other human beings. The process of propagating the information has evolve over the history from the creation of the first human language, passing through the invention of writing, up to more recent propagation mechanism based on technology innovations such as mass media communication (e.g., radio, TV, etc.). Researchers in different areas have always been interested in answering questions like how, when or how fast the information is propagated. For example, we can find relatively old studies digging into this intriguing issue in fields like traditional media communication [G64] or social science [BR87]. Furthermore, the irruption of the Internet have brought the modern society to the so called Information Era in which human beings have access to a huge volume of information as it never happened before in the History. This trend has been multiplied by the recent irruption of OSNs that have rapidly become one of the most used information propagation media for hundreds of millions of people. Therefore, the described context has defined the understanding of the information propagation in OSNs as a topic of great relevance for the scientific community that have performed initial characterization studies of information propagation in some popular OSNs like Twitter (TW) [KLP+10], Facebook [SRM09].

In this research we focus on characterizing the propagation information in a new OSN, Google+ (G+). In G+, similarly to other networks such as Facebook, the basic piece of information is the so called post. A post can attach different types of content (e.g., a simple text, a video, a photo, etc.). The process to propagate information in Google+ occurs as follow: First, the post is initially fed into the system by a user that we refer to as root user. From this moment the post is available in the G+ wall of the root user and it is accessible either to all users in G+ (if the root user defines it as a public post) or to a limited number of users selected by the root user (e.g., his work colleagues). Any G+ user with access to the post can reshare it, which makes that post available in that user’s wall. Then, the post is exposed to a new set of users that in turn could decide to also reshare the post. Therefore, each post in G+ generates a propagation tree (or reshare tree) that constitutes the basic information propagation structure that we use for our analysis. In addition, a piece of external content to the OSN (e.g., an URL) can be posted by multiple root users within the system, each of them generating an individual propagation tree. These trees form a meta-structure that indeed represents the propagation of that piece of external content in the OSN. We refer to this meta-structure as propagation forest.

To the best of the authors' knowledge, there is any previous large-scale study of the properties of the propagation forests for a major OSN.

In particular, the main goal of this research is the characterization of the main properties of the propagation trees and forests in G+. To achieve this goal, we have developed a sophisticated crawling tool that allowed us to collect all public posts available in the system and related information to each of them like the total number of reshares and the type of post (e.g., text, video, photo, etc.). Overall, we collected 540M of posts since the release date of G+ (June 28th 2011) during a period of two years (until July 3rd 2013). Next, we leverage a public feature of G+ named Ripples [VWB+13] that provides the reshare tree of each post that has been reshared at least once and the reshare forest associate with each external content that has been shared through an URL in G+. In addition, the Ripple of a post (or external content) provides detailed information such as the timestamp or the user-id associated to each reshare. Our final dataset includes almost 30M reshare trees after filtering those activities without reshares and more than 34M reshare forests. We will leverage this data to carefully characterize the propagation trees and forests in G+ as follows.

In the first part of the research we focus on the analysis of the main properties of reshare trees in G+. Our analysis addresses the following questions: which volume of posts is actually propagated in G+?, how many users typically propagate a post in G+?, how far and how fast a post travels in G+? To this end we study the main spatial and temporal properties associated to each one of the 30M resharers trees in our dataset. First, we study the spatial properties of the reshare trees. This is, what are the size and the height of the reshare trees in G+ that permit us to characterize the volume of posts that propagates, the number of users that typically propagate a post and how far posts travel in G+. Second, to characterize how fast information travels in G+, we use a temporal metric named root delay that measures the time difference between the original posting time and the time of each reshare in the associated tree. Finally, we compare the results obtained for the analysis of the spatial and temporal metrics with those obtained for TW by two different sources: our own dataset including information of more than 2.3M tweets and the results reported in [KLP+10]. To the best of the author's knowledge, our comparative analysis is the largest-scale study comparing the information propagation between two major OSNs performed so far.

In the second part of this research we focus on the characterization of the reshare forests. In this case we address the next questions: what is the typical number of root users for a given external content? Does the most popular content propagates through a large number of small trees or through few very large trees? For this purpose, we again characterize the main spatial and temporal properties of the propagation forests.

Finally, to conclude our study of the information propagation in G+, we analyze the importance that the social links established in the OSN has for the content dissemination; in other words, whether posts are typically propagated between users that have previously established a social link or, on the contrary, information propagation follows different paths to those defined by the social connectivity graph.

In summary, this research presents four main contributions that extend the existing work: (i) We present the first characterization study of information propagation in Google+ (G+) using all the publicly available information in the system; (ii) To the best of our knowledge, this research presents the largest-scale head-to-head comparison of information propagation between two major OSNs, Google+ and Twitter; (iii) We present the first large-scale study that characterize the propagation forests in a major OSN; (iv) We also present (to the best of the authors knowledge) the first analysis of the importance that the social connectivity graph has in the information propagation.

The analyses conducted in the research led to the following main findings:

- We confirm that only a minor fraction of the information published in OSNs is propagated. This indicates that most of the information posted in major OSNs is not interesting enough for anyone to share it.
- Although the information is propagated faster in Twitter than in G+, it gets more reshares and travels longer paths in G+. Furthermore, the probability of getting a post reshared is higher in G+ than in TW. This is a side effect of the selective way in which G+ shows the content to the users rather than the sequential way followed by Twitter.
- Most of the popular external content in G+ present forests formed by a large number of small trees rather than few big trees. Moreover, the lifespan of external content through their associated forests is significantly longer than the lifespan of posts through their associated trees.
- Around two thirds of posts propagate between users that do not have a social connection established. It seems that hot topics and communities plays an important role in this phenomenon since they enable a user to reshare posts made by users with whom the former has not established social connections. Furthermore, our analysis reveals that information travels faster through existing social links. This seems to be due to the fact that the information posted by a user appears immediately in its followers' walls, but a third user would need to access to the hot topics or to the specific community to see the post and reshare it what logically takes more time.

3.1.1 Measurement Methodology & Datasets

The activity unit in G+, similarly to Facebook, is the post. When a user publishes a post, his followers can forward (i.e., propagate) that post to their respective followers by means of a reshare. The followers of the followers can also reshare the post and so on. Then, an original post along with all its reshares can be organized in a tree that we refer to as reshare tree or propagation tree. Moreover, the posts published can contain external content as a Youtube video or a link to an online newspaper. All the propagation trees referring to the same external content can be grouped in propagation forests.

By analyzing the main properties of the reshare tree associated to a large number of posts in G+ we can characterize the information propagation in G+. Moreover, the analysis of the propagation forests allows us to understand the importance of external factors (as the external popularity of the content) in the propagation of the information in the network.

In this section we describe our measurement methodology to collect all public posts and its public reshares in G+ that form the basic data to conduct our characterization study. Furthermore, we also present the filtering techniques used to process the collected data.

3.1.1.1 Measurement Methodology

Our aim is to collect all posts in G+ and its associated reshare trees as well as the propagation forests associated to a given external content. To this end, we follow a methodology divided in 3 phases.

In the first phase we leverage the technique described in our previous work [GCR+13] to capture the ids of all users connected to the Largest Connected Component (LCC) of G+ in July 2013.

Using as input the collected list of user ids, in the second phase, we leverage the G+ API to collect all public posts of each user in the LCC of G+. Since the G+ API limits the number of queries that can be done from a single IP address to 10K per day, we use a distributed architecture of proxies installed in PlanetLab nodes and more than 600 G+ accounts in order to speed up our data collection process. In particular, using our tool we collected every single (public) post published by every user within the LCC of G+ from the release date of G+ (June 28th 2011) until the date in which we started this second phase (July 3rd 2013). Our crawler needed 72 days to collect 540M (public) posts. For each post the crawler retrieved the following information of interest for this research: total number of reshares (including both public and private reshares) and type of post (e.g., photo, video, text, etc.).

Finally, in the third phase, we leverage a public feature of G+ named Ripples [VWB+13]. Each public post reshared at least once in G+ has an associated Ripple page in which the reshare tree associated to the post is available including relevant information such as the id and the language (if available) of each user, the timestamp of each reshare and the parent-child relationships within the reshare tree. We also leverage this G+ feature in order to obtain the propagation forest associated to each external content published in G+

We use a web crawler that retrieves the previous information for the reshare trees associated to each public post (with at least one reshare) obtained in the second phase.

Using the previous methodology we obtain a dataset formed by 29.6M reshare trees that overall include 90M nodes. We refer to this dataset as G+ reshares. Furthermore, our G+ forests dataset is composed for 34.7M propagation forests referring to a content that has been shared more than one time in the Online Social Network. All these propagation forests together are composed by more than 615M nodes. Finally, we want to clarify that both the original posts and the reshares collected with our tool are public since neither the G+ API nor the Ripples provide information about private posts or reshares.

3.1.1.2 Dataset Filtering

In a first manual inspection of our dataset we discovered the presence of an important fraction of large reshare trees in which the original post and most of the reshares were done by the same user. In some cases, the same user reshared its post more than 1K times. We suspect that these users are bots (an example of such users can be found at https://plus.google.com/u/0/112555830876915762462).

The goal of our research is to characterize the information propagation in G+ and thus if a user reshares its own post no propagation event occurs. Then, we filter all links in which the parent and child are the same user by merging both nodes in a single one in the propagation tree.

3.1.1.3 Other Datasets

In order to compare the main characteristics of the information propagation in G+ and TW, we have collected a dataset including the number of retweets for 2.3M tweets collected from more than 17K randomly selected users. We refer to this dataset as TW-retweets. This dataset was collected between March 28th 2013 and April 2nd 2013. Furthermore, part of our comparison analysis with TW will refer to the results obtained by Kwak et al. [KLP+10] using a dataset collected in 2009 (3 years after the release of TW).

3.1.2 Basic Characterization of Information Propagation in G+

Our goal in this section is to characterize the information propagation in G+. To this end, we analyze a set of spatial and temporal properties along with other metrics associated to the propagation trees in our G+ reshares dataset. Furthermore, in order to put our results into a meaningful context we compare them with those reported for TW in [KLP+10] or obtained from our TW-reshares dataset.

3.1.2.1 Fraction of Propagated Information

The first step to characterize the information propagation in an OSN is to understand what fraction of the available information in the system actually propagates. To this end, we have computed the percentage of posts (tweets) in our G+ reshares (TW reshares) dataset that have at least 1 reshare (retweet). The results indicate that just a small fraction of posts is propagated in both networks. In particular, only 6.8% and 3.3% of the posts/tweets are reshared in G+ and TW, respectively. However, despite both percentages are small, it is important to highlight that the probability of getting a post reshared in G+ is roughly double than in TW. We conjecture that this is due to the fact that the overall volume of activity is over an order of magnitude larger in TW than in G+ [GCR+13] and then TW presents a much longer tail of non-propagated tweets that leads to the reported result.

3.1.2.2 Public vs. Private Information Propagation

In Twitter most of the available information is public due to its broadcasting nature [KLP+10]. However, in G+ (similar to FB) users can set up different privacy configurations and decide whether their posts are public (available to anyone) or private (accessible just to some selected users). An early study revealed that around 30% of the posts published in G+ are public [KBH+12].

We can accurately compute the percentage of public reshares for each post within our G+ reshares dataset. As indicated in Section 3.1.1, the G+ API provides the total number of reshares (private and public) for each post whereas the Ripples functionality only reports the public reshares. Then, we can divide the number of public reshares by the number of total reshares to obtain the fraction of public reshares for each post in our dataset.

Our results indicate that, overall, 51% of the reshares in our dataset are public. This suggests that roughly half of the propagated information in G+ is disseminated in a public way. Furthermore, Figure 66 presents the CDF of the percentage of public reshares for the posts in our G+ reshares dataset. In particular, we consider three groups of posts for our analysis: All represents all posts that have at least 1 reshare in our dataset; +10 includes all posts that have at least 10 reshares in our dataset (i.e., mid-popular posts); +100 has all posts that have at least 100 reshares in our dataset (i.e., popular posts). For All we observe that more than 50% of posts have either 0 or 100% public reshares. Most of these are posts with just a single reshare that can be either private or public. In addition, as we increase the popularity (i.e., number of reshares) of the group of posts under consideration there is a reduction in the fraction of public reshares. This suggests that popular posts tend to keep a larger fraction of their propagation trees private.

Note that, unless otherwise stated, in the rest of the document we analyze the public part of the propagation trees associated to the posts within our dataset. Therefore, most of our results refer to the propagation of public information in G+, that as reported above represents roughly half of the whole public propagated information.

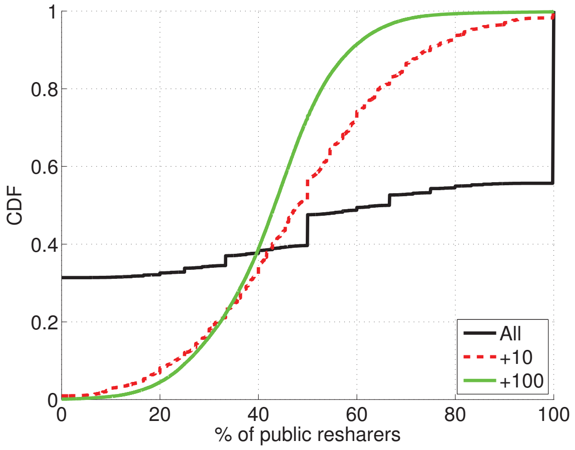


Figure 66. CDF of percentage of public resharers per post. We plot the results for three set of posts grouped according to the number of reshares they attract.

3.1.2.3 Spatial Properties of Propagation Trees in G+

In this subsection we study two spatial properties that are essential to properly characterize the information propagation phenomenon in G+:

-Tree Size is defined as the total number of nodes that form the propagation tree of a post. This is, the original post (we refer to this node as root node) and all the reshares. This metric captures the popularity of a post.

-Tree Height is defined as the number of levels forming the longest branch of a tree. This metric captures how far the information travels from the root node.

3.1.2.3.1 Tree Size Analysis

Let us start by analyzing the distribution of the size of propagation trees in G+. To this end Figure 67 shows the CDF of the size for the propagation trees within our G+ reshares dataset. In particular we consider the following groups of posts: All includes all posts with at least 1 reshare in our dataset; All-Public includes all posts in our G+ reshares dataset with at least 1 public reshare; +10-Public includes the posts in our G+ reshares dataset with at least 10 public reshares (mid-popular posts); +100-Public includes the posts in our G+ reshares dataset with at least 100 public reshares (popular posts). Furthermore, the figure presents the distribution of the size for the propagation trees of tweets with at least 1 retweet in our TW-retweets dataset. We refer to this group as TW-All.

The results indicate that 90% of the trees have a size ≤ 5 and ≤ 6 for All-Public and All, respectively. Surprisingly, this value is 3 in the case of All-TW. Finally, there is not any remarkable observation to mention for +10-Public or +100-Public.

Therefore, our results indicate that the propagated information attracts more reshares in G+ than in TW.

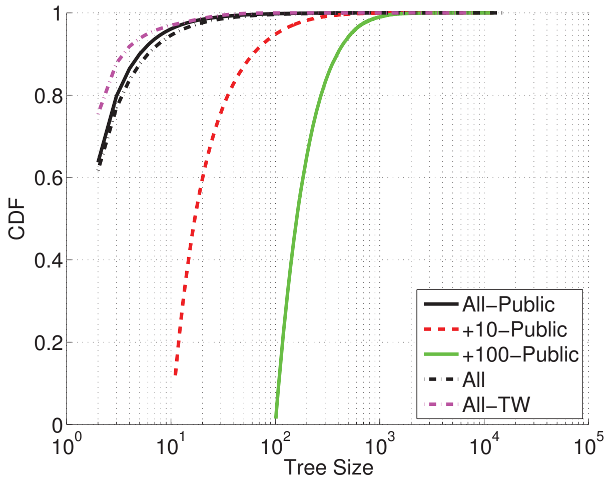


Figure 67: CDF of the tree size per post for different groups of posts within our G+ reshares and TW retweets datasets

3.1.2.3.2 Tree Height Analysis

Now we focus in analyzing the height for the propagation trees in G+. Figure 68 presents the CDF of the tree height for the propagation trees in our G+ reshares dataset. In this case we only have information for the groups of posts including public reshares (All-Public, +10-Public and +100-Public). Furthermore, our TW retweets dataset does not include information about the height of the propagation trees. Then, we refer to the results obtained by Kwak et al. [KLP10+] for the comparison with TW.

We observe that 6.8% of All-Public trees have a height ≥ 1 in G+ in front of the 3.3% reported for TW. Furthermore, it is interesting to notice that the highest tree in our G+ dataset presents 129 levels whereas Kwak et al. [KLP10+] report a maximum height equal to 11 for Twitter3.

In short, our results indicate that information travels longer paths in G+ than in TW.

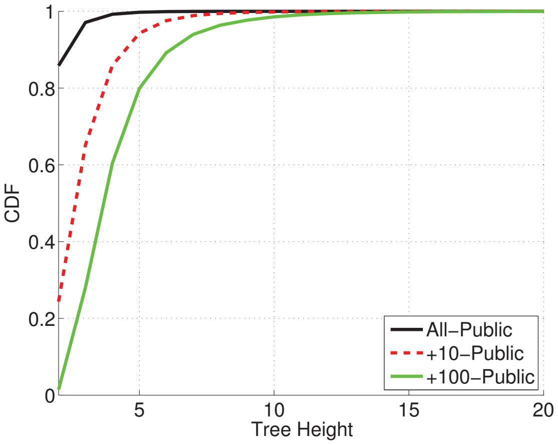


Figure 68: CDF of tree height for different groups of posts within our G+ reshares dataset

3.1.2.4 Temporal Properties of Propagation Trees in G+

The aim of this subsection is to analyze the temporal properties of the propagation trees. To this end we define the Root Delay as the time elapsed between the instant a node reshares a post and the original posting time. This metric captures the overall propagation delay of a post across the entire reshare tree.

Note that, as it occurred for the case of the tree height, we can only obtain the value of this temporal metric for the public reshares in our dataset and then we present the results for All-Public, +10-Public and +100-Public. Furthermore, our TW-retweets dataset does not include information regarding these temporal metrics. Thus we will refer to the results reported by Kwak et al. [KLP10+] for the comparison between G+ and TW, as we did for the discussion of trees' height.

Figure 69 shows the distribution of the root delay for nodes in All-Public, +10-Public and +100-Public. The results show that 80% of all public reshares happen in the first 24 hours after the original post was published and the median root delay is equal to 4.4 hours. Furthermore, Kwak et al. report a median root delay lower than 1 hour for Twitter. These results are a side effect of the way in which each system shows the information to the users, sequentially in the case of Twitter and selectively in the case of G+

Hence, we conclude that information propagates faster in Twitter than in G+.

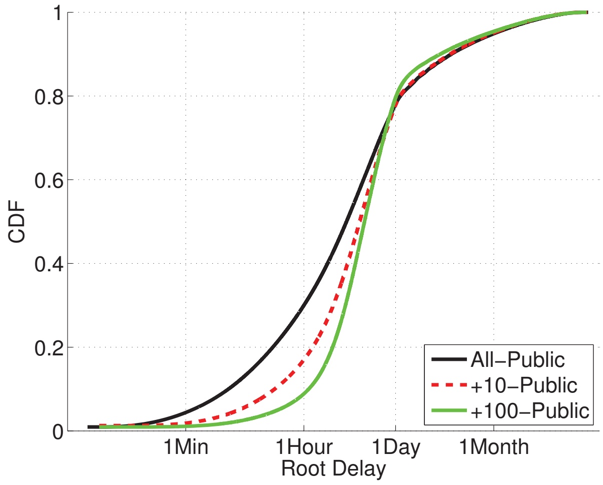


Figure 69: CDF of root delay. We plot the results for three sets of public posts grouped according to the number of public reshares they attract

3.1.2.5 Users Participation in Resharers Trees

The contribution of different users to the information propagation in major OSNs such as Twitter has been reported to be skewed [CHB+10]. Indeed, the capacity of a user to disseminate information is dictated by its influence. In this section we study the skewness in the contribution of users to the information propagation in G+.

In order to conduct our analysis we rely on a metric that we refer to as Reach (R). The Reach of user u in a tree t is computed as the number of nodes in t located in the subtree below u. If u is the root node, then R(u, t) is equal to the tree size - 1.

Using this basic concept we define two metrics that capture two different types of user's influence:

- Total Reach (TR): This metric is computed as the sum of the Reach of user u across all the propagation trees in which u participates. The formal expression of T R for a user u that has participated in T trees is as follows:

$$TR(u) = \sum_{t=1}^{T} R(u,t) \qquad (1)$$

- Avg. Reach (R): This metric is computed as the average Reach of user u across all the propagation trees in which u participates (including those original posts from u without reshares). The formal expression of R for a user u that has participated in T trees is as follows:

$$\overline{R} = \frac{1}{T}\sum_{t=1}^{T} R(u,t) \qquad (2)$$

In Figure 70 we present a graphical example with two propagation trees in order to further clarify the introduced metrics. We compute the Reach for nodes A, B, C and D in both trees and present it beside these nodes. In addition, we include a table at the bottom of the figure that shows the Total Reach and Average Reach for those nodes.

The Total Reach and the Avg Reach present complementary versions of a user's influence. On the one hand, TR of a user u captures the aggregate number of people that u has reached with all his posts and reshares. Thus, it measures the overall capacity of a user to propagate information. On the other hand, the R measure the capacity of the user to attract attention with each one of his posts.

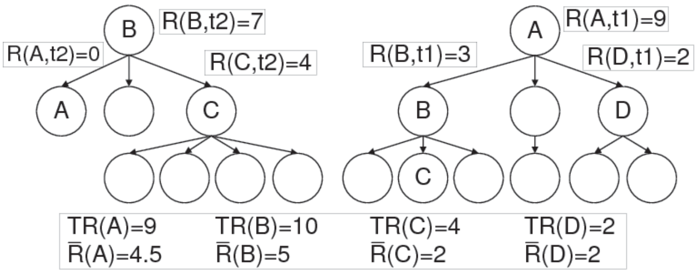


Figure 70: Graphical example to explain the metrics Reach, Total Reach (TR), Avg. Reach (R) using two propagation trees

We start our analysis by studying the contribution of different users to the propagation of information in G+. A skewed contribution would reveal the presence of influential users. In particular, we study the distribution for the two defined metrics, TR and R.

Figure 71 shows the portion of reshares included in our dataset (y-axis) associated to a given percentage of users (x-axis). In other words, it depicts the skewness of the distribution of Total Reach across G+ users. We observe that there are few users (1%) with a very high Total Reach that concentrate most of the reshares (85%).

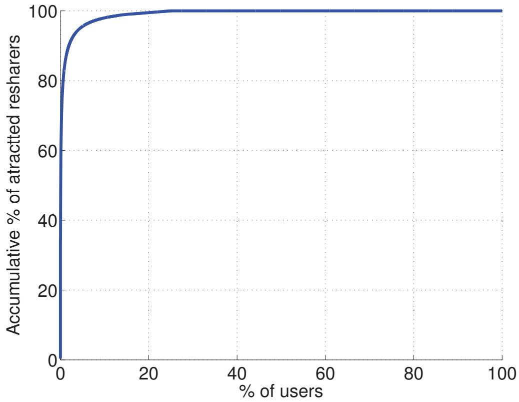


Figure 71: Skewness of the total reach across G+ users

Figure 72 presents the CDF of the Avg. Reach across G+ users. We observe that just 140 and 31 users present an R ≥50 and ≥100, respectively. Then, we confirm that only a very low fraction of users is able to systematically attract a large number of reshares for their posts.

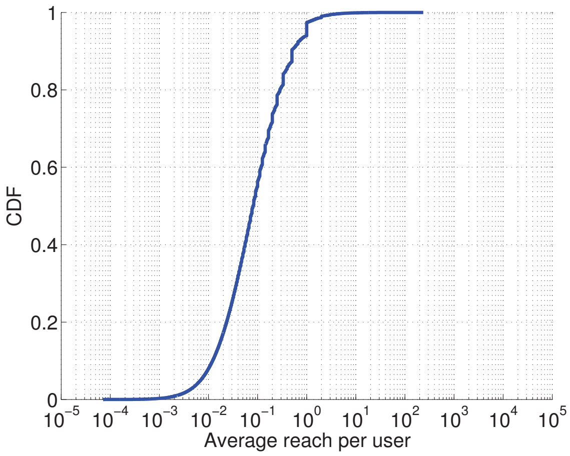


Figure 72: CDF of the average reach for G+ users

3.1.2.6 Summary

In this section we have characterized the main properties of information propagation in G+ and compared them with those of another major OSN such as TW. The main outcomes of our analysis are:

- A common characteristic of propagation information in major OSNs is that a very small fraction (< 7%) of the information available in these systems propagates. This provides an indicator of the fraction of interesting information available in major OSNs.
- The information propagates faster but follows shorter paths in Twitter than in G+. This is a consequence of the way in which information is shown in each system. Sequential-based systems such as Twitter force short-term conversations among their users whereas Selective-based systems such as those used in G+ or Facebook chooses which content to show to each user based on his preferences, volume of interactions with other users, etc. This helps to prolong the lifespan of conversations in the OSN.
- The higher popularity of a post in G+ translates into a longer lifespan of that post in the system. As future work, it would be interesting to analyze this aspect in other major OSNs in order to confirm if this is a common property of major OSNs. This property differentiates G+ (and possibly other OSNs) from other popular applications in the Internet (e.g., p2p file-sharing) in which popularity is mapped into flash-crowd events.
- We observe a very skewed distribution for the Total Reach and Average Reach among the G+ users. It indicates only a few users are attracting the attention of the network.

3.1.3 Basic Characterization of Content Propagation in G+

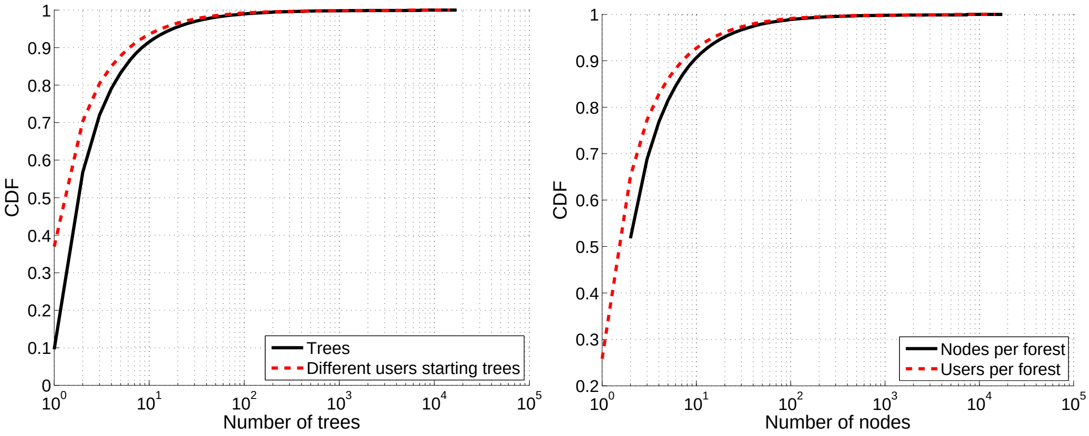
In this section we characterize how a given external content propagates on the network. For this purpose we analyze some spatial and temporal properties of the propagation forests associated to external URLs in G+.

3.1.3.1 Spatial Properties of Propagation Forests in G+

In this section we study two of the main spatial properties of the propagation forests in G+: -Number of trees per forest is the number of times a content has been originally posted in the social network. This variable give us an intuition of the external content popularity, since the social network activity does not affect to the number of times an external content is originally posted.

-Forest size is the number of nodes forming the forest. This includes the original posts and all the reshares generated by any of them. This metric captures the popularity of a given content inside the social network.

It is worthy to remark from the more than 113M out of the 148M forests in our dataset have been shared by a single users who have not attracted any reshare. We do not take these ”single node” forests into account for the following analysis.



(a) CDF of the number of trees per forest

(b) CDF of the forest size

Figure 73: Forest composition

3.1.3.1.1 Number of Trees per Forest

Figure 73(a) shows the CDF for the number of trees per forest and the number of different root users generating these trees. We can observe only 5% of the forests contains a unique tree. It suggest that the propagation of a given content depends on external factors since if a content is popular enough to attract the attention of more than one person it usually will be originally shared more than once. Moreover, only 6.7% of the forests have more than 10 trees and only 0.8% of them have more than 100. Finally, the widest forest includes more than 10K trees.

Focusing now in the number of users who originally share the same content we can observe for about 30% of the forests a unique user has originated all the reshared trees. Since we have previously observe that only a 15% of the forests contains only one tree it seems that at least for 15% of the forests under study a single user is sharing more than one time the same content. Nevertheless as in the case of the number of trees we can find forests in which more than 10K users are originally sharing the same content.

3.1.3.1.2 Forest Size

Figure 73(b) present the CDF of the number of nodes per forest as well as the CDF for the number of different users participating in each forest. We observe that almost 55% of the nodes are composed for only two nodes. Moreover, the distribution is very similar to the distribution obtained for the number of trees. This indicates that the number of trees and nodes per forest is usually very similar, thus, we conjecture that forests are typically composed by very small trees. We validate this hypothesis in the next subsection.

3.1.3.1.3 Forest Size vs. Number of Trees

To validate the above hypothesis in which we stated that forests are typically formed by very small trees we carefully analyze the relation between the forest size and the number of trees in the forest.

Figure 74 presents the boxplot for the average tree size dividing the population under study in different groups using the size of the tree. The results in any case give us an average tree size between 1 and 1.25. This result confirms our previous hypothesis: the forests are usually composed by small trees.

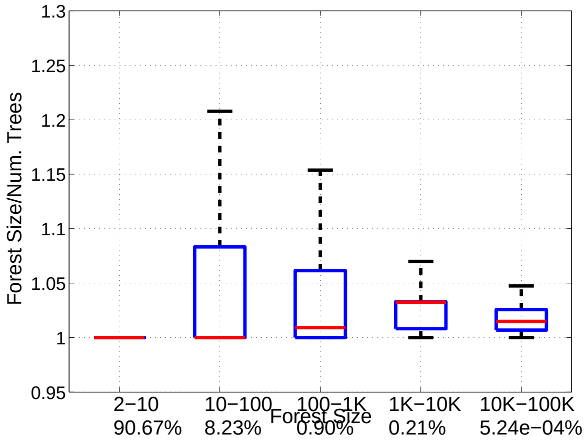


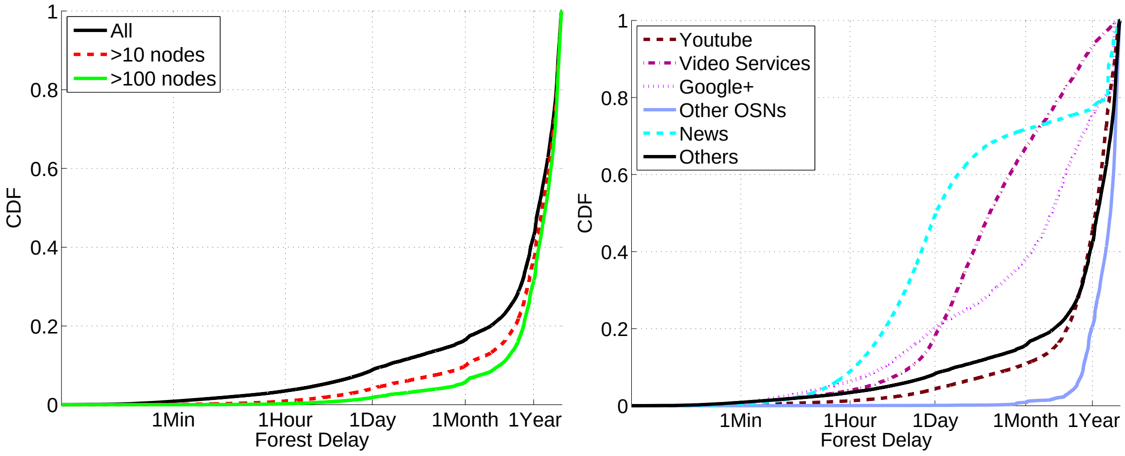
Figure 74: Average tree size per forest

3.1.3.2 Temporal Properties of Propagation Forests in G+

As we did in Section 3.1.2, after analyzing the spatial properties of forests we now focus on studying their temporal properties. In this case we analyze the Forest Delay, a metric analogue to the Root Delay. This metric is defined as the time elapsed between the instant when a user share or reshare a content and the instant when this content was posted for the first time. This metric captures the overall propagation delay of a content across the entire reshare forest.

Figure 75(a) shows the CDF of the forest delay for all the nodes across our dataset. Contrary to the case of the root delay, where 80% of the reshares happened during the first 24 hours, for the forest delays we obtain higher delays. We can observe that after 1 month just 20% of the reshares have been performed. While an internal content (e.g. a Photo) posted by a user will be available in his followers' wall during a limited period of time and external content (e.g., a YouTube video) can be accessed by a user and posted in G+ at any moment. This explains the shorter lifespan of propagation trees compared to propagation forests. Indeed, we observe that half of the reshares associated to a forest typically happen 1 year after the content was originally shared by a user in G+. This finding is of high interests in disciplines like marketing since it indicates that the popularity of, for instance, a publicity spot may peak in OSNs not at the release of it but several months after.

Finally, we would like to understand the temporal properties of the forest generated by different type of external content. For this purpose we use the domain of the URL to identify the type of associated content. Figure 75(b) shows the CDF for the forest delay for each one of the defined categories. We observe that for news webpages like cnn.com or nytimes.com 50% of the propagation is made during the first day. While this result is far from the observations made in the previous section for a single post it indicates that this kind of content tend to be consumed in the first hours after it is generated. Finally, we observe two unexpected results: first, the lifespan of videos shared in Youtube is much longer than the lifespan of videos shared in other services like vimeo.com or dailymotion.com and second, a careful analysis of the forests for OSN domains reveals that these long lifespan is not due to the continuous active presence of the link during the reported lifespan. Instead, we observe that the trees associated to other OSN links are typically very far apart in time. Finally, the links to other G+ pages propagates faster than the links to other external OSNs.



(a) Divided by forest size

(b) Divided by content type

Figure 75: CDF of the forest delay

3.1.3.3 Users Participation in Resharers Forests

This section analyzes the behaviour of the users who have participated in the propagation forests in our dataset. For this purpose and following the methodology used in the previous section we analyze three variables:

-The Number of times a user has participated in the forests indicates the activity level of a given user. In this sense, it is also important to analyze whether users post the same content several times or if they tend to share content that usually come from the same domain.

-The User average reach (R) and total reach (TR) as defined in Section 3.1.3.3.

3.1.3.3.1 How Much the Users Post?

Figure 76 presents the CDF of the number of times the same user appears in our G+ forests dataset, the number of different contents this user has shared in the system and from how many different domains this content comes. For this purpose, we use as domain the hostname of the shared URL removing the subdomains starting with m. or www. (i.e., m.youtube.com and www.youtube.com count as youtube.com while play.google.com and feedproxy.google.com count as different domains).

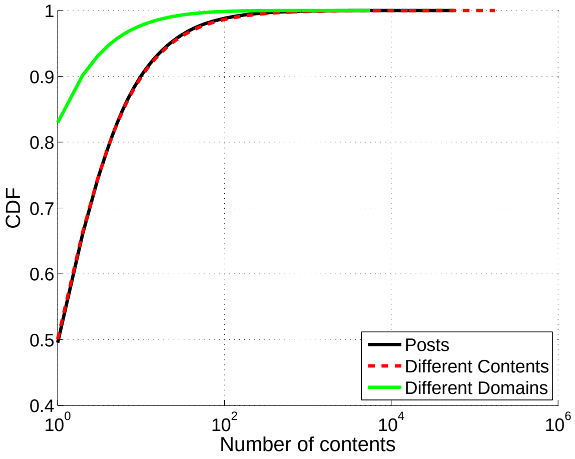


Figure 76: CDF of the number of contents posted per user

We observe that half of the users participates only once in the propagation Forests in our dataset. Moreover, the distribution of the number of times and of the number of different contents are very similar. It indicates that users usually share each content only once. To explain why we observed in the previous section a non-negligible number of trees where a user shares the same content several times we should focus in the tail of the distribution. A manual inspection of these users allows us to identify some accounts that belongs to robots/spammers that automatically post a huge amount of links several times. As an example the user https://plus.google.com/106373251267437926474/ appears more than 8M times in our dataset, but it shares less than 52K different URLs coming from less than 220 different domains.

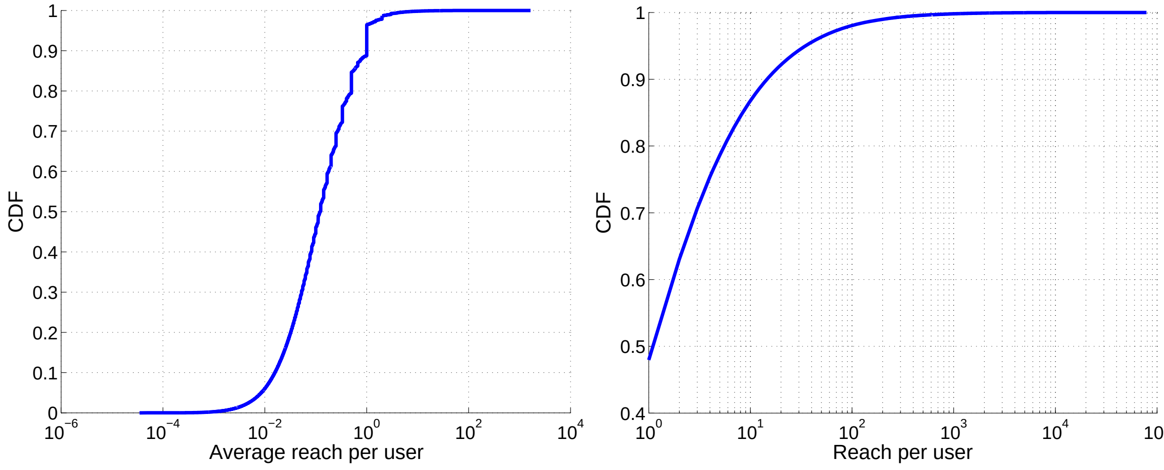
Moreover, we can observe the number of domains from where the information comes is smaller than the number of different URLs. More than 80% of the users post content from only one domain. While the effect of the aforementioned robots/spammers is important in this metric, it is also worth to mention that youtube.com is the most popular domain accounting for 37.8% of the total reshares and 18% of the different URLs in our dataset while the second one, ow.ly, only represents 1% of the reshares, 2% of the different URLs and 0.2% of the users.

3.1.3.3.2 Capacity of the Users to Attract Resharers

Our previous analyses demonstrate the existence of a major portion of trees with a single node inside the studied forest. This suggests that it is very difficult for standard G+ users to obtain reshares of their posts.

First, it is important to remark that only 1.3M (3.43%) out of the 37M of users participating in the Forests in our dataset have obtained at least one reshare across the forests where they have participated. Figure 77 presents the CDF of the R and TR for these 1.3M users receiving at least 1 reshare. The R for 88.75% of the users is smaller than 1. This indicates that these users receive less than one reshare per post. Nevertheless, there is a small number of users (5.3K) receiving more than 5 reshares per publication.

Finally, we can observe that the value of TR is equal to 1 for about 50% of the users who only manage to obtain one reshare among all their posts. In this case only 2% of the users have attracted more than 100 resharers summing up all their posts.



(a) Avg. reach

(b) Total reach

Figure 77: CDF of the avg. and total reach per user

3.1.3.4 Summary

In this section we have characterized the main properties of the propagation forests in G+. The main outcomes of our analysis are:

- The propagation forests in Google+ are composed basically of small trees. It suggests external factors as the content popularity have a key importance in order to understand the content propagation inside a social network.
- In contrast to previous studies and the section 3.1.3, the lifespan of a content inside the social network is usually very long and it depends on the kind of content.
- Standard users does not usually share the same content more than once and it is easy to identify a non-negligible number of robots/spammer by only identifying users who post the same content several times. Nevertheless it is common for standard users to post different URLs belonging to the same domain (i.e., youtube.com).

3.1.4 Mapping of Connectivity Graph and Propagation Trees

To conclude our analysis, in this section we study whether the public information in G+ is propagated following the social connectivity graph or it follows other paths. To this end, we check whether each link in the propagation trees in our dataset exists in the social connectivity graph of G+. Surprisingly, only 34.4% of the observed reshares happen between a user and one of his followers.

Figure 78 presents a boxplot for the percentage of reshares received per user that have been done by the user's followers. The users have been divided in different groups depending on the number of different users from which they have received reactions. Intuitively, we can expect non popular users will have a strong dependency on the social graph in order to disseminate the information posted, nevertheless, if we consider users with a single reshare, only 25% of them receive this reshare from a follower. Moreover, when the users receive a higher number of reshares the percentage of their followers resharing them decreases. A manual inspection of the post receiving this non-follower resharers suggest the effect of the Google+ communities and the Hot Topics as a possible cause for this non expected results.

To further understand the role played by the social graph in the propagation of the information Figure 79(a) presents the CDF of the average time needed to reshare a post from another user, depending on whether there exists a relation or not. We can observe that the propagation is faster during the first hours when the relation between the users exists. This is an expected result since any user can see his friends' activities in his own wall just after the publication has been posted. However, when the relation does not exist the information propagates faster after the first day. This effect can be caused by the time needed for a post to become a hot topic or due to the difference in the access time of the users to the G+ community page.

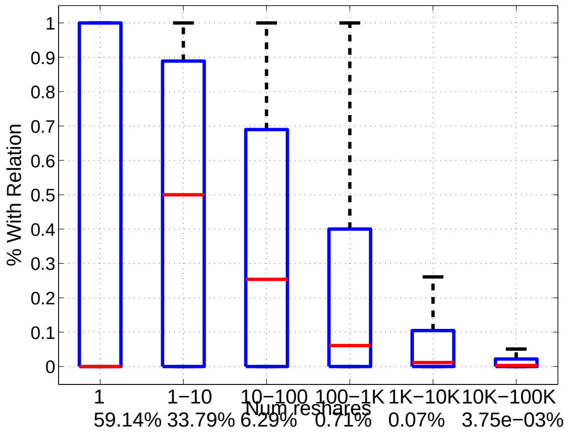
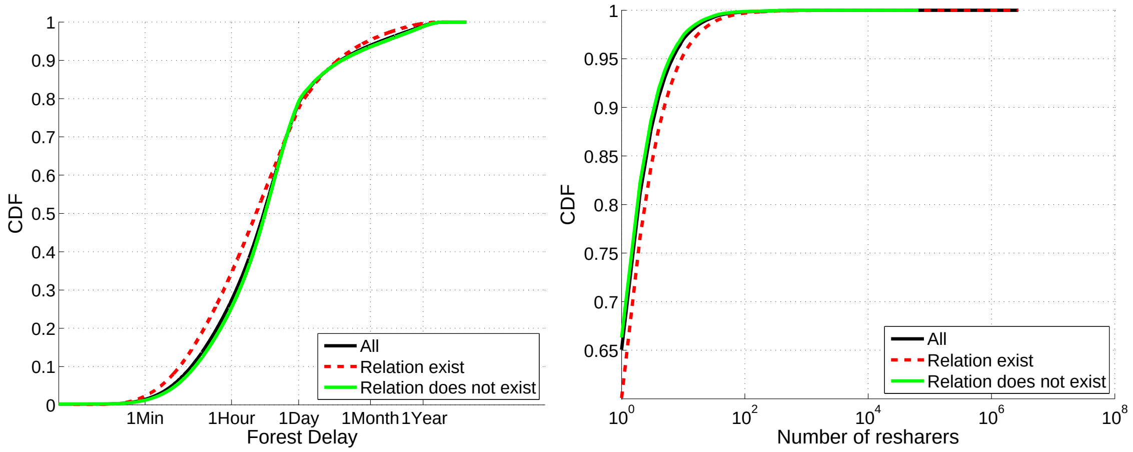


Figure 78: % of resharers made by the followers of the user

Finally, we analyze the number of reshares associated to a link between two users. For example, if user A has reshared 4 contents published by user B, we will assign a value of 4 to this metric. Figure 79(b) shows the CDF of this metric. We observe that the reshares are more frequent when the social link exists. Nevertheless, the difference observed is smaller than we could expect. For example, when the social link exists, 40% of the times the users reshare more than one content from the same users whereas this percentage only decreases to 35% if the social link does not exist.



(a) Average delay in the reshare

(b) Num resharers per relation

Figure 79: Effect of the social graph in the propagation of the content

3.1.4.1 Summary

In this section we have studied how the social graph maps the content propagation in G+. The main findings are:

- Only one third of the reshares happen between users without a social connection established. Moreover, this percentage decreases for popular users attracting a bigger number of reshares.
- The role played by the social graph in the content propagation is important during the first hours after the content has been published. Nevertheless, after this initial phase other factors, such as the G+ Hot Topics or the communities, are more important.

3.1.5 Related Work

OSN characterization. The successful irruption of social networks in our daily life has attracted the attention of the research community in the last years. We can find several works that characterize the main properties of the most popular OSNs such as Facebook [BBR+12, UKB11] or Twitter [HRW08, KLP+10]. More interesting for our research, there exists a number of efforts that are focused on the characterization of Google+. The first works in this area studied G+ graph properties [SSS+12, MCS+12]. More recent research has made a step further and has investigated the evolution of the graph and the activity of users in G+ [GCR+13, GHM+12]. Finally, there are some studies that have focused on concrete aspects like the new circle feature introduced in Google+ [KBH+12, FFL12] or the study of collaborative privacy management solutions [HAJ12].

Information propagation characterization. The propagation of information and the factors that influence it have been historically studied in areas like social science [BR87] or traditional media communication [G64]. However, these studies were usually limited to a small population. The irruption of technologies like Web 2.0 and OSNs, which allow hundreds of millions of users to interact among them every day, have allowed extending the information propagation studies to a much larger population. Therefore, in the recent years, we have experienced an increasing proliferation of works that address the information propagation in areas like viral marketing [KKT03, LAH07], Internet blogs [10] or systems like Arxiv [8]. In addition, we can also find several studies that analyze the propagation of the information in social networks like Flickr [CAA+09, CAG09, YF09], Twitter [KLP+10, CHB+10, YW10], Digg [LG10] or Facebook [SRM09]. Finally, some works exploit the knowledge extracted from previous studies and propose novel solutions that, for instance, improve the performance of OSNs [SMM+11, SYC08] or define a system to quickly detect natural disasters [SOM10].

Our research presents four main contributions in the area of information propagation in OSNs. (i) This is the first effort to characterize information propagation in G+. (ii) Previous studies have just analyzed the information propagation in a sample of the OSN. In contrast, our work considers all the information that is publicly propagated in G+. (iii) To the best of our knowledge, this research is the most extensive head to head comparison in the information propagation between two major OSNs, G+ and Twitter. (iv) Finally, this work presents the first large-scale characterization of the propagation of external content in a major OSN.

3.1.6 Conclusions

This research characterizes the propagation of the information in one of the major OSNs, Google+, using all the public available information in the system including 540M posts, 30M propagation trees and 34M propagation forests. We present the largest-scale head to head comparison of the information propagation between two major OSNs, Twitter and G+. The comparison has revealed that a standard post is disseminated quicker in Twitter, but it attracts more reshares and travels longer paths in G+. Furthermore, the probability of getting a post reshared is higher in G+ than in TW. These results are a side effect of the way in which each system shows the information to the users, sequentially in the case of Twitter and selectively in the case of G+.

In addition, this research presents the first large-scale characterization of the external content propagation through propagation forests in a major OSN. The analysis reveals that external popular content is typically shared in a large number of small trees instead of few big trees. Moreover, the results indicate that the lifespan of an external content within the OSN through multiple trees reaches usually months or even years and depends on the type of external content.

Finally, we have also analyzed the influence of the social connectivity graph in the propagation of information in G+. We conclude that around two thirds of the reshares happen between users that do not have an established social connection. This suggests that other factors, such as the Hot Topics or the G+ communities, in addition to the social graph, play a key role in the information propagation in G+.

Overall, the extensive conducted analysis poses us in a better position to understand the basis of information propagation in G+, in particular, and OSNs, in general.

3.2 Macroscopic Geographical Information Propagation Analysis in Twitter

Since the existence of online social media, citizens around the world use it to communicate beyond media blackouts. For example, IRC channels served as a way for individuals to report news in 1991 during the media blocks in the Soviet union putsch and in the Gulf War [CAGW]. The growth of social media use in developed societies allowed individuals to take one step further, organizing actions and spreading relevant news around their environment. One example of such emergence of coordination away from mass media are the reports and actions taken by anonymous users against the Church of Scientology in 2008 [BMH+11], which they claimed to manipulate mass media channels. More recently, the widespread adoption of social media around the world has triggered events that were reported by individuals beyond media blockages, including actions of social movements like the Spanish ”Indignados” [BRG+11], the Gezi protests in Turkey [T14], and the revolutions during the Arab Spring [Z14].

While social media is clearly relevant in news reporting nowadays, there are still many open questions about their potential, their role in informing the population, and their limitations and inherent biases. Social media overcome some limitations of traditional mass media that are commonly attributed as sources of biases. First, the cost to set up an information channel in social media is negligible, allowing individual users to become news channels themselves. This overcomes the ownership barrier of news in traditional media [HC08] and potentially weaken biases related to information centralization. Second, social media have the potential of a very broad and deep coverage of all kinds of news, allowing any news piece to be found, reported, and eventually attract collective attention under the right circumstances. On the other hand, social media are not free of the influence of other biases that can limit their transparency and coverage. For example, a major part of the funding in social media comes from advertising strategies, in which the product is the attention of users and not the reported news [H13]. Furthermore, social media are not isolated communities, and traditional mass media biases are likely to resonate in each social medium.

The selection and content of news media can be affected by subjective and/or systemic biases. Subjective biases operate at the level of the individual information of the news reporter, during the evaluation of the informativeness of a news piece in the context of current events [GR65]. In that evaluation of what is newsworthy and what is not, additional subjective factors can bias individual choice, including beliefs, information overload [GGS14], and cultural preferences [GT13]. Systemic biases operate at a mesoscopic level, creating patterns that cannot be observed at the level of a news piece or a journalist, but can be observed at larger scales when sufficient content is analyzed [HC07]. In the context of international news, there is no evidence that can attribute these biases to supranational power structures [W98], but incentive mechanisms can bias news if journalists and news media are subject to economic and social forces [O65, HC08]. Examples of empirically tested presence of these systemic biases relate them to increase reporting with GDP of the country where news originate [I96], and decreases with geographical distance [CSB87] and political stability [KA14].

International factors in mass media have been an active field of research for the last decades, but works on news in social media mainly focused on individual selection [SCL13] and political biases [AQC+14, AQC], as well as collective dynamics [CHP+14] and emotional reactions [CMM14]. Our aim is to study the international structure of news in social media, testing for the existence of systemic biases related to economic factors and for the role of subjective biases with respect to cultural similarity. News are defined as selected information on current events, shared through a communication channel that allows a group or a society to access them as they happen [S08]. One of the main roles of news media is to select what is relevant or newsworthy, filtering out information that is not of interest for their audiences. While the data sources for mass media news are trivially defined by newspapers, radio and television channels, news in social media can aggregate in Facebook discussion groups, link sharing communities like Reddit [SFM+14], or aggregation mechanisms in microblogging platforms. In this research, we focus on Twitter Trending Topics (TTs), which serve as a global filtering mechanism for Twitter users to define in a collective manner which information is relevant and when. This way, TTs serve as a centralized channel of communication from many to many, constituting news due to their self-organized public interest and their strong time component and global reach through the Twitter interface. TTs serve as a contrast with the traditional view of mass media in which information flows to many from very few, and those few are the ones who unilaterally decide what is newsworthy.

From a methodological perspective this research presents two mayor contributions:

(1) We analyze the international coverage of hundreds of thousands of TTs across tens of countries in 2013 and 2014. Applying state-of-the-art community detection techniques and some basic network theory concepts on these datasets we derive the international structure of TTs coverage. Moreover, we analyze the underlying demographic, economical and cultural biases of the unveiled structure.

(2) We develop a novel methodology, that leverages the Google News service, to classify TTs into external (news appearing in mass media that are also reported in Twitter in the form of TTs) and internal (not reported by mass media).

Moreover, the main findings resulting from our analysis can be summarized as follows:

(1) The international coverage of TTs inherits some demographic and economical biases from mass media since the flow of TTs also follows the gradient of wealth from big richer to poorer countries. However, contrary to mass media, cultural similarity is the dominant factor that determines the flow of TTs between countries.

(2) Similarities between the international coverage of TTs in Twitter and news in mass media lead to a large overlapping in the specific news covered by both media. Moreover, the mentioned fundamental differences make Twitter an alternative media that distributes, across countries, news not reported in mass media.

(3) The comparison of the surging date of thousands of overlapping news in mass media and Twitter (as TTs) reveals that Twitter is, typically, ahead than mass media in news reporting.

3.2.1 Methodology & Datasets

In this section we describe our measurement methodology to collect thousands of Local TTs across tens of countries over several months and to present the datasets that we use for our analysis in the rest of the research.

Twitter provides different APIs to access the information available in the system [TWAPI]. In our methodology we leverage the REST API that, among other information, provides the list of 10 TTs at a given instant and for a given location (e.g., a country). We query the Twitter API from a created Twitter application. However, Twitter imposes a maximum rate of queries per application, then in order to speed up the data collection process we created multiple Twitter applications to query the API. This parallel crawling technique allows us to collect the list of Local TTs for each country every 5 minutes that is the interval used by Twitter to update the list of TTs in a given location [TTAPI].

We have collected two datasets using the measurement methodology described above. The first one includes 112K Local TTs from 35 countries and was collected over a period of 3 months between February 20, 2013 and May 20, 2013. The second one includes 188K Local TTs from 62 countries1 and was collected over a period of roughly 3 month between April 14, 2014 and July 04, 2014. We refer to these two datasets as TT-2013 and TT-2014, respectively. Other Datasets: Finally, to complete our analysis we use two complementary datasets: (i) the World Bank database (year 2010) [TWB] from where we extract different demographic and economic related metrics such as the GDP of a country or the trade cost and immigration flow between a pair of countries; (ii) The Hofstede cultural dimension dataset [LBH80], which quantifies the culture of various countries into dimensions of shared values, from which we use the 4 principal ones: Power Distance, Individualism, Masculinity, and Uncertainty Avoidance.

3.2.2 Detecting Leader-Follower Relationships

The temporal sequence of appearance of Local TTs allows us to analyze the structure of leader-follower relationships among countries in Twitter. This type of relationship constitutes a media bias in which the temporal patterns of news are a manifestation of an alignment of incentives, rather than a causal relationship. Herman and Chomsky detail this kind of bias in their Propaganda model [HC08], operationalizing it as a sourcing filter in which some sources are overlooked, distorting the information presented to the public. Thus, leader-follower relationships can appear without a hidden power that manipulates media outlets in different countries; they can be the product of a set of shared interests that create ordered patterns where news originate and where they are consumed afterwards. After all, timing in trending news is critical, and details of news might not be as important as reporting them the first.

We detect leader-follower relationships among countries through the ordering of the appearance of Local TTs. If such relationship exists, there will be a tendency for Local TTs to appear in the leader country before they emerge in the follower country. Thus, we take into account in our analysis the events in which a Local TT appears in country Ci at time ti, and later in country Cj at time tj > ti. To test if these temporal sequences are not the product of independent events unrelated to leader-follower relationships, we apply the model of priority processes and bursty patterns in queue theory [OB05, VOD+06]. If Local TTs appear in pairs of countries by chance, as the result of independent phenomena, the time intervals between the appearance Δt will follow an exponential distribution P(Δt) ~ λe−λΔt as the result of decoupled Poisson processes [VOD+06]. On the other hand, if the appearances of Local TTs follow communication channels in which countries take TTs from other countries, the correlations present in the time sequence will make the distribution of Δt follow a power-law P (Δt) ~ Δt−α [VOD+06, WZX+10]. This kind of temporal patterns are known to appear in communication processes, including the correspondence of Einstein and Darwin [OB05], e-mail communication [VOD+06], mobile phone messaging [WZX+10], chatroom interaction [GGS+12], and Twitter dialogues [GWR+14].

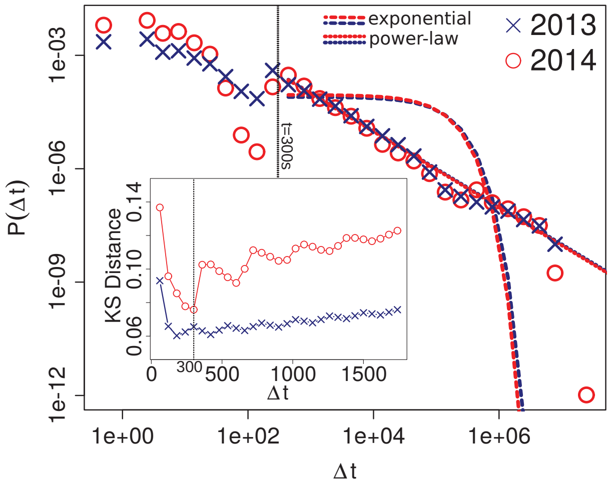


Figure 80: Distribution of time intervals between the appearance of TT for both datasets, including power-law and exponential fits. Inset: KS Distance between the empirical distribution and a power-law fit for ranging values of the minimum Δt of the fit.

Figure 80 shows the distributions of times between the appearance of TTs in different countries P(Δt), for both TT-2013 and TT-2014 datasets. To test the alternative hypotheses of the existence or nonexistence of correlations between TTs appearances, we fit power-law and exponential distributions through maximum likelihood on the Kolmogorov-Smirnov criterion [CRN08, ABP14]. The resulting theoretical distributions are plotted in Figure 80, suggesting that a power-law fit is better than the exponential alternative. Log likelihood ratio tests [ABP14] between the empirical data and the power-law models give significant estimates of 198.22 (TT-2013) and 293.03 (TT-2014), providing very strong support to reject the independent events hypothesis, in favour of the hypothesis that the appearance of TTs follows a correlated process in the leader-follower relationships.

The exponent of the fits are αTT-2013 = 1.00005 ± 10−6 and αTT-2014 =1.004±10−5, indicating that the process of following TTs is saturated and has a finite queue [VOD+06], i.e. countries have strong limitations in their capacity to adopt TTs in comparison to how many are generated in the rest of the world. However, note that these correlations are not observable at all timescales, as the power-law distributions are fit above a minimum value of Δt. The inset of Figure 80 shows the Kolmogorov-Smirnov estimate (KS) for a set of minimum values, revealing that the minimum for TT-2013 and TT-2014 are 3 and 5 minutes, respectively. This means that correlations at a timescale smaller than 3 minutes cannot be observed in any of our datasets, and that between 3 and 5 minutes the results are inconsistent. Thus, we take 5 minutes as a minimum criterion to deduce the manifestation of a leader-follower event. This coincides with our sampling frequency, allowing us to remove from the data those co-occurrences for which we do not have sufficient evidence to consider them product of a leader-follower process.

We quantify the tendency for country Cf to follow county Cl through the amount of TTs that appeared in Cf at least 5 minutes after they appeared in Cl. When applied to every pair of countries in our dataset, these counts define a weighted, directed network in which nodes are countries and links have weights corresponding to the number of TTs that appeared in the leader-follower relationship. We refer to this graph as the international structure of TTs coverage

Some further considerations about our model are:

- We refer to the countries in which a given TT appears in the first Δt (i.e., 5 minutes) after the surge of a TT as sources since based on our model they cannot be followers of any other country for this TT. Then, source countries seem to participate in the generation of TTs.
- Our model captures well the appearance sequence between countries located in different time zones. For instance, a TT that surges in a western European country during the early morning and appears, after few hours, in a Latin American country during its early morning represents a leader-follower relationship between the two involved countries. Our model would accurately identifies this as a precedence event between the European and the Latin American country.

3.2.3 Analysis of Individual Countries' Properties

In this section we first analyze the capacity of each individual country to generate local (i.e., non-shared) and internationally shared TTs. Afterwards, we focus on shared TTs in order to understand whether the sharing activity of a country concentrates in few or, contrary, spreads across a large number of other countries. We present the results for the TT-2014 dataset in this section.

Generation of non-shared Trending Topics: Figure 81 shows the percentages of TTs that (i) remain local within a country (i.e., are not shared with other countries) and (ii) are shared with other countries. Moreover, the figure presents the total number of TTs associated to each country (in the right y-axis). First, we observe that a large fraction of TTs corresponds to local events in a country that do not attract enough attention among users abroad to become TTs in other countries. Indeed, half of countries in our dataset have more than 50% non-shared TTs. Second, some countries such as Japan, Korea and Turkey present an extreme localization with 99, 96 and 95% of non-shared TTs, respectively. This leads them to an almost isolated condition by which they exchange a negligible volume of TTs with other countries. The use of non-Latin alphabets and special characters by these three countries is a plausible explanation for their observed isolation.

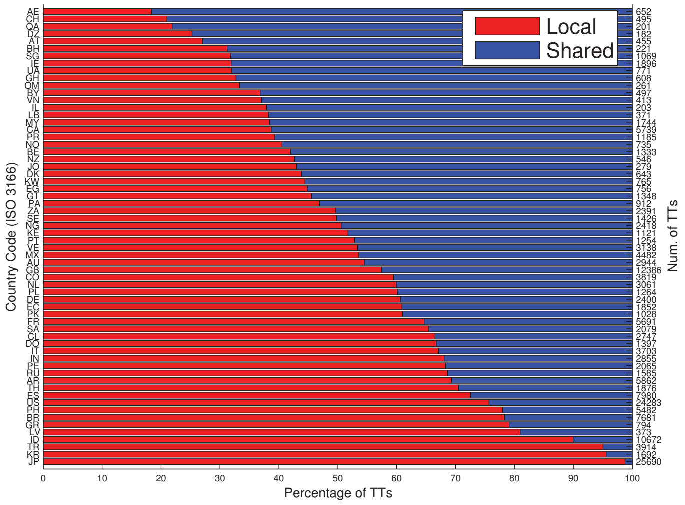


Figure 81: Fraction of local and shared TTs (x-axis) and total number of TTs (right y-axis) for countries (represented by their country code in the left y-axis) in our TT-2014

Generation of shared Trending Topics: A first indication of the contribution of individual countries to the international sharing of TTs is given by their overall number of shared TTs. Figure 81 provides this information as the product of the total number of TTs and the fraction of shared TTs. US and GB share a significant larger number of TTs than any other country. Specifically, they share 67% and 50% more TTs than the third largest contributor (Canada), respectively. This is a first indication of the important role that these two countries may have.

However, a more critical aspect to define the importance of a country is its capacity to generate TTs that are afterwards consumed by others. The right-y axis in Figure 82 presents the number of TTs that has been generated in each country, according to our model in Section 3.1.3 these are TTs for which the country is a source. The results show that US and GB are the countries generating a larger number of TTs. In this case the gap with the third contributor (Canada) grows up to 333% and 232% for US and GB, respectively. This confirms the key role of US and GB in the international sharing of TTs.

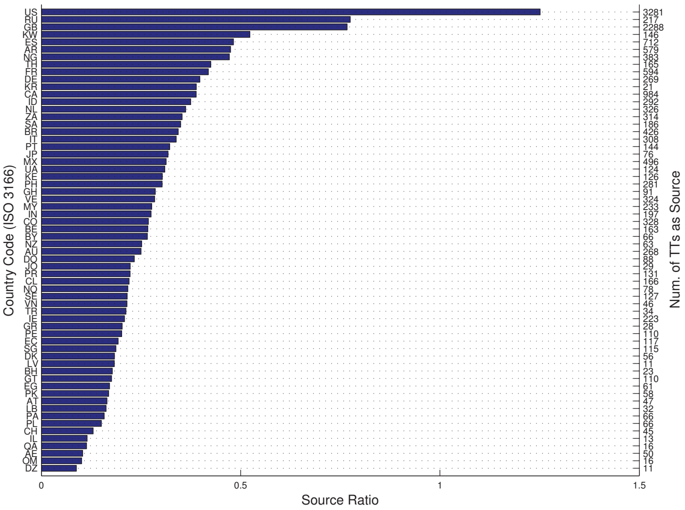


Figure 82: Source Ration (x-axis) and total number of generated TTs (right y-axis) for countries (represented by their country code in the left y-axis) in our TT-2014

Finally, we study the bias of countries towards generating or consuming TTs. To this end we compute the Source Ratio (SR) as the ratio between the number of TTs in which the country acts as a source and the number of TTs in which it is not a source but a consumer. Figure 82 presents the SR for each country in the x-axis. We observe that all but four countries (US, GB, Kuwait and Russia) present a SR < 0.5 and indicating that they consume more than twice TTs than they generate. This result is a clear sign of centralization in which few countries generate a large fraction of TTs internationally consumed. In addition, it is worth reporting that US is the sole country presenting a generator profile, indeed, it generates 25% more TTs than it consumes. Surprisingly, GB presents a SR equal 0.75, then, despite its important contribution of generated TTs, its volume of consumed TTs is even higher.

Dispersion of countries' sharing activity: To conclude the analysis of individual countries properties, in this subsection we analyze the distribution of shared TTs of a country across other countries in order to understand whether the attention of a country concentrates in few other countries (i.e., share TTs with few other countries) or, contrary, it disperses across many other countries. Note that our model defines bidirectional relationships between countries, and then we study the dispersion from the perspective of both leading and following activity of a country.

To compute the dispersion of the attention for the leading activity of a country Ci, we calculate the number of TTs for which each other country has followed Ci. Then we compute the Gini Coefficient across these samples. The Gini Coefficient is a measure of statistical dispersion commonly used to measure inequality. It varies between 0 (complete equality) and 1 (complete inequality). In our case, a small (high) Gini coefficient indicates dispersion (concentration) in the attention of Ci's followers.

Moreover to compute the dispersion of the attention for the following activity of a country Ci, we calculate the number of TTs in which Ci follows each other country. Then, we compute the Gini coefficient across these samples to conclude whether the attention of Ci is concentrated (dispersed) in few (across many) other countries.

The Gini coefficient is a relative metric and then we need to provide some context to properly interpret the obtained results. To this end, we have calculated the leading and following Gini coefficients for the nodes of a random Erdos-Renyi graph with the same properties (number of nodes, number of links and overall weight) as our empirical graph.

Figure 83 shows the leading and following Gini coefficients for each of the countries in our dataset and for the equivalent nodes in the correspondent random graph. Note that the horizontal lines represent the median values of the Gini coefficient for each case. In particular, the median leading (following) Gini coefficient is 5 (3) times larger in the empirical graph than in the correspondent random graph. This shows an important level of concentration in both the attention attracted by countries from others and the attention countries dedicate to others. Moreover, the attention dedicated to others is concentrated in fewer countries than the attention received.

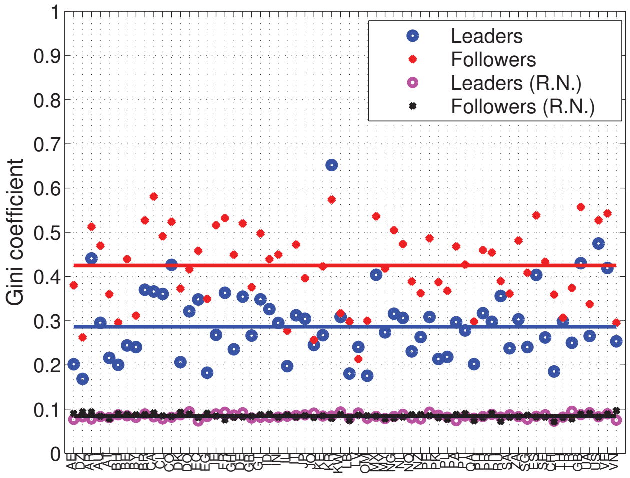


Figure 83: Leading and following Gini coefficients for the graph of countries derived from our TT-2014 dataset and its equivalent random graph

In summary, the conducted analyses of the properties of individual countries reveal some initial insights about the international structure of TTs coverage: (i) An important fraction of TTs are associated to internal events in a country that do not attract much interest abroad and thus become TT only in that country; (ii) There are few isolated countries that share a negligible fraction of TTs with other countries; (iii) There seems to be a large heterogeneity in the contribution of different countries to the international structure of TTs coverage. While US and GB generate a large volume of internationally shared TTs, most other countries present a strong consumer profile; (iv) Most countries receive the attention from few other countries and concentrate their attention in an ever smaller group of other countries. The last observation suggests that international structure of TTs coverage may be formed by communities of countries with stronger socio-economical ties that present a higher sharing activity among them than with others. We analyze this hypothesis in detail in the next section.

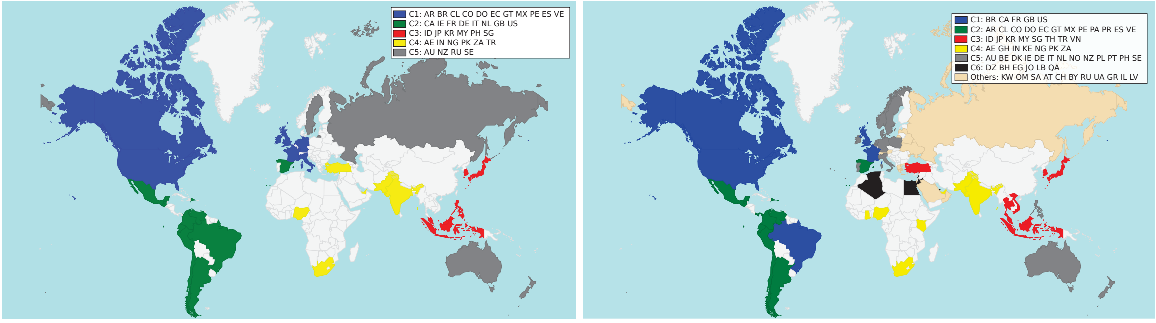
3.2.4 International Structure of TTs Coverage

The analysis of individual countries' properties conducted in the previous section provided initial evidences about some key aspects of the international structure of TTs coverage such as the existence of different communities of countries with similar interests and the presence of countries (US and GB) with a very relevant role in the sharing of TTs. In this section, we leverage the weighted directed graph representation of our TT-2013 and TT-2014 datasets to apply community detection techniques and network theory metrics to confirm the two previous observations. Indeed, the application of these techniques will allow to accurately characterize the international structure of TTs coverage. Moreover, we will study demographic, economical and cultural biases of the unveiled structure. Finally, we present a brief qualitative comparison of the international structure of TTs coverage and its counterpart for traditional news reported in the literature in order to discuss the differences on the international sharing of news through mass media and social media as well as the underneath biases of such phenomena.

3.2.4.1 Identifying Main Components of the Structure

Identifying Community Structure: We identified the communities in the graphs of TT-2013 and TT-2014 datasets through the Q-modularity for weighted, directed graphs [ADF+07, GJA09]. To find the optimal partition into communities, we used the fast heuristic method of radatools with 1000 bootstraps, ensuring an optimal solution thanks to the reduced size of the networks under analysis.

Once we got a partition, we applied the same method to each of the communities, to analyze if they can be further divided in a hierarchical way. Figure 84 presents the main communities in a world map for our TT-2013 and TT-2014 datasets.


🟦 C1: AR BR CL CO DO EC GT MX PE ES VE
🟩 C2: CA IE FR DE IT NL GB US
🟥 C3: ID JP KR MY PH SG
🟨 C4: AE IN NG PK ZA TR
⬛ C5: AU NZ RU SE
🟦 C1: BR CA FR GB US
🟩 C2: AR CL CO DO EC GT MX PE PA PR ES VE
🟥 C3: ID JP KR MY SG TH TR VN
🟨 C4: AE GH IN KE NG PK ZA
⬛ C5: AU BE DK IE DE IT NL NO NZ PL PT PH SE
⬛ C6: DZ BH EG JO LB QA
🟫 Others: KW OM SA AT CH BY RU UA GR IL LV

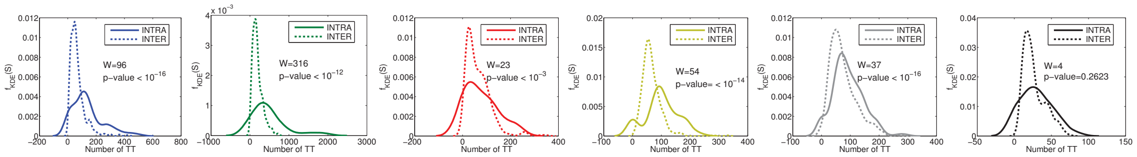
(a) TT-2013

(b) TT-2014

Figure 84: Representation of main communities in a world map for TT-2013 and TT-2014 datasets. We recommend to see this figure in color.

The 35 countries in TT-2013 form 5 different communities: Community 1 formed by the Latin-America countries and Spain. This community includes all the Spanish-speaking countries in our dataset; Community 2 formed by US, CA and European countries; Community 3 formed by East-Asian countries; Community 4 formed by West Asian and African Countries; Community 5 formed by Australia, New Zealand, Russia and Sweden. All these communities seem to include countries with strong cultural, economical and geographical ties, excepting Community 5 in which the relation between AU and NZ with SE and RU is not clear.

Our TT-2014 dataset includes 62 countries and thus the number of communities increases to 10. The 4 relevant communities reported for 2013 are still present in 2014, with some slight differences that we describe next: (i) Community 1, losses Brazil and incorporates new countries from Central America and the Caribbean such as Paraguay or Puerto Rico. With the lose of Brazil this community is formed exclusively by the Spanish Speaking countries in our dataset; (ii) Community 2 in 2013 is split into two communities, 2 and 5, in 2014. Community 2 is formed by US and CA and two European countries (GB and FR) with strong ties to US and CA. Finally, BR joins this community that overall is formed by countries where Twitter has a very high penetration [CGC+14, RCC+13]. Community 5 is mainly formed by the majority of European countries in our dataset; (iii) Community 3 and 4 incorporate few new countries from their respective geographical areas, e.g., Community 3 incorporates Thailand and Vietnam whereas Community 4 incorporates Ghana and Kenya. Therefore, it seems that the international structure of TTs coverage in 2014 is a natural evolution from the one in 2013. Moreover, among the new communities we find a mid-size one formed mainly by Middle East countries (Community 6), whereas the rest are small communities formed by 2 or 3 countries with low representativeness.



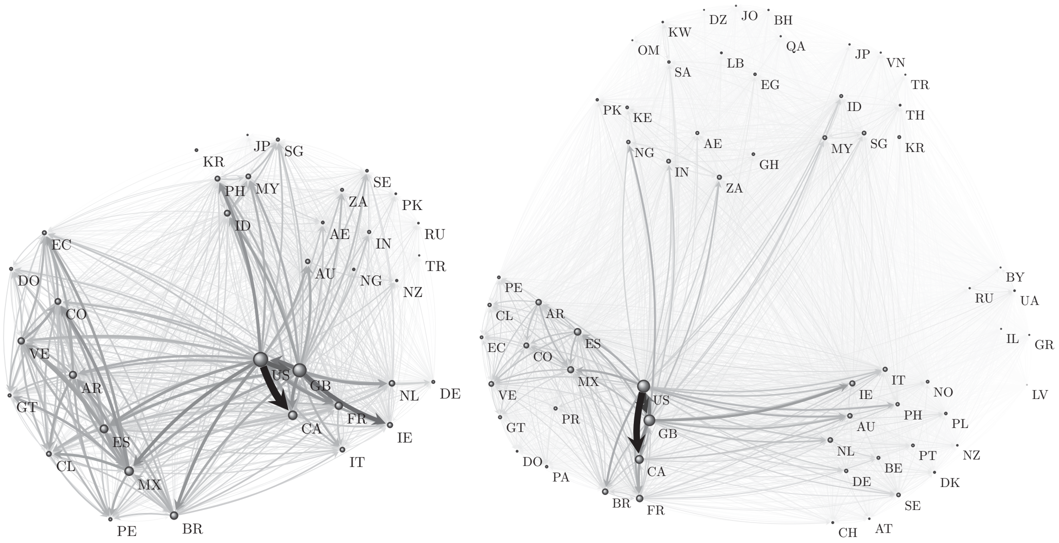
(a) Comm. 1 (b) Comm. 2 (c) Comm. 3 (d) Comm. 4 (e) Comm. 5 (f) Comm. 6

Figure 85: Kernel density probabilities (KDP) for the intra- and inter-community connectivity distribution for the communities including more than 4 countries in TT-2014 dataset and W and p-value for the Wilcoxon test on the KDP distributions for each community

To evaluate the statistical significance of the most relevant unveiled communities we have computed the intra- and inter-community cohesion for those communities with more than four members in our TT-2014 dataset using a Kernel Density Estimation (KDE) method [HTF01], where cohesion is measured based on the weights of the leader-follower links. Figure 85 shows the obtained results. In addition, for each of these communities, we have conducted a Wilcoxon test [W45] on the distributions of intra- and inter-community edge weights. The Wilcoxon test indicates whether the two compared distributions have the same average (null hypothesis) without assuming normality, and its point estimate W measures the difference between the medians of the compared distributions. Therefore, a positive value of W indicates a stronger internal than external cohesion for a community. In addition, the larger W is the stronger the intra-community cohesion is. Figure 85 presents the W and p-values obtained from the Wilcoxon test for each analyzed community. The test rejects the null hypothesis in all cases, and thus the intra- and inter-community similarity distributions are statistically different (p-value< 0.05) for all communities excepting community 6. Moreover, the internal cohesion of these communities is higher than its external cohesion. In particular, communities 1 and 2 show the strongest internal cohesion with W values equal to 96 and 316, respectively. Finally, note that initially Community 3 was not statistically significant (p-value > 0.05). The cause was the presence of two strongly isolated countries (JP and KR) as members of this community. The reported values in Figure 85 are obtained after removing JP and KR from the community.

Identifying main country hubs in the structure: After identifying and characterizing the main communities present in the international structure of TTs coverage, we focus on evaluating the relevance of individual countries in this structure. To this end, we compute different centrality metrics for each country in the graph representation of such structure. The centrality metrics are commonly used in graph theory to identify important nodes in a graph. In particular, we have computed the out degree (i.e., the number of relationships for which a country is a leader) and the betweenness centralities. The obtained results confirm that US and GB are the main communication hubs in the international structure of TTs coverage. For instance, US (GB) show an out-degree and a betweenness centrality at least 52% (36%) and 523% (469%) larger than any other country in our TT-2014 dataset, respectively.

Finally, to conclude this section, Figure 86 shows the graphical representation of the unveiled international structure of TTs coverage for 2013 and 2014 as a weighted, directed graph using the cuttlefish visualization software [CBW] and the ARF layout algorithm [G07]. The size and darkness of each link are proportional to its weight whereas the size of each node is proportional to its out-degree. Moreover, the location of nodes is arranged according to attractive and repulsive forces, with additional attraction that clusters nodes in the same communities.



(a) TT-2013

(b) TT-2014

Figure 86: Visual representation of the international structure of TTs coverage for TT-2013 and TT-2014 datasets

3.2.4.2 Bias Factors of the Structure

In this subsection we analyze the socio-economical biases that have driven the generation of the reported structure of TTs coverage across countries. In particular we first consider demographic and economical biases and afterwards we study the existence of cultural biases.

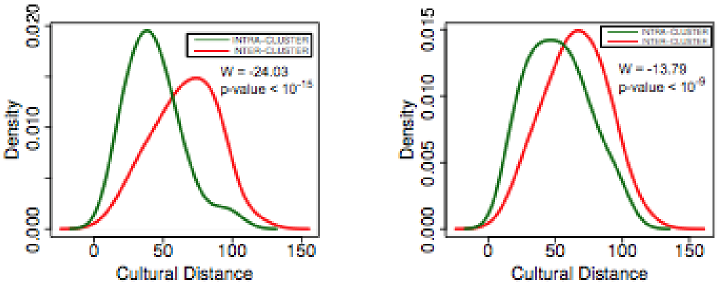
Demographic and Economical Biases: Combining data from the TTs datasets with World Bank statistics allows us to test the existence of demographic and economical biases in the international structure of TTs coverage. To do so, we apply a linear regression model to all pairs of countries (C1, C2), in which the dependent variable is the rank of the amount of TTs in the leader-follower relationship where C1 leads and C2 follows. The rank transformation allows us to cope with the skewness of the link weight distribution, calculating this way Spearman type correlations through our linear regression. Note that this implies a non-linear relation of the variables, and effect sizes cannot be interpreted in the scale of amount of TTs, but on the scale of positions in the ranking. The independent variables, obtained from the World Bank database, are the amount of migrants from country C1 to country C2 (migration1), the amount of migrants in the reverse direction during the same period (migration2), the trade costs between both countries, and the difference in GDP (GDP1-GDP2).

The results of this model (referred to as model 1) are shown in the first rows of Table 19. Migration in both ways is significant, but the effect size of migration from C1 to C2 is larger, showing that there is a strong tendency for immigrants of a country to give relevance to the TTs appearing in their country of origin. The difference in GDP is significant in both datasets, with an effect size of 14 (2013) and 34 (2014) positions in the rank of TTs per trillion USD in GDP differences. This suggests that TTs follow the gradient of wealth: TTs created in rich and big countries are more likely to appear in follower countries with less wealth and power. This bias has been also reported for the coverage of traditional news over mass media [W93, W02]. In the presence of GDP and migration statistics, trade cost is not significant, showing that the phenomenon of TTs biases is not related to the state of economic relations among countries but on their inequalities and demographics. Finally, all effect sizes are stronger and p-values smaller in 2014 than in 2013, indicating that the extension of our analysis to more countries reveals clearer biases.

Year	model	migration1	migration2	trade cost	GDP difference	Cultural distance
2013	1	**0.00011(0.0012)**	**0.00007(0.027)**	−0.0009(0.103)	**$1.487 \cdot 10^{-11}(< 10^{-8})$**	-
2014	1	**$0.00056(< 10^{-8})$**	**$0.00048(< 10^{-5})$**	−0.0013(0.098)	**$3.451 \cdot 10^{-11}(< 10^{-8})$**	-
2013	2	**0.00008(0.0093)**	0.00006(0.067)	−0.0003(0.785)	**$1.390 \cdot 10^{-11}(< 10^{-5})$**	**−1.445(0.00222)**
2014	2	**0.00035(0.0008)**	**0.00032(0.002)**	−0.0024(0.087)	**$2.655 \cdot 10^{-11}(0.0004)$**	**$-4.772(< 10^{-4})$**

Table 19: Regression weights and p-values for models 1 and 2 in both TT-2013 and TT-2014. Significant values ($p < 0.05$) are reported in boldface

Cultural Bias: A simple inspection of the members of the main reported communities (see Figure 84) suggests the existence of a clear cultural bias in the formation of the international structure of TTs coverage. In order to test this claim we compute the Hofstede's cultural distance between each pair of countries in our two datasets. Note that we calculate the cultural distance between two countries as the Euclidean distance in the space of Hofstede's cultural dimensions. To test the role of culture in the presence of demographic and economic biases, we extend the linear regression model adding cultural distance as an independent variable (we refer to this model as model 2). The result is shown in the last rows of Table 1, revealing that cultural distance has a negative weight on the amount of TTs shared from one country to another, and thus countries closer in cultural space are more likely to follow each other. It is worth to notice that adding this effect of cultural distance weakens the effect sizes of other variables, where migration and GDP differences remain significant. This dominance of cultural biases is a differentiating factor with respect to the international coverage of traditional news by mass media [W02, W93].



(a) TT-2013 (b) TT-2014

Figure 87: KDPs and W and p-values from the Wilcoxon test for the intra- and inter-community cultural distances

In addition, to test if the reported cultural bias is expressed within the communities of the structure of TTs coverage, we group the cultural distances into: intra-community distances between countries belonging to the same community and inter-community distances between countries belonging to different communities. Then, we compute the Kernel Density Probability for the intra- and inter-community distances in our TT-2013 and TT-2014 datasets and run a Wilcoxon test to compare both distributions. Figure 87 presents the obtained results. The Wilcoxon test reveals that distributions of inter- and intra-community distances are statistically different in both TT-2013 and TT-2014. Moreover, the intra-community cultural distance is clearly smaller than the inter-community distance, confirming the existence of cultural bias within the reported communities.

Finally, the presence of a strong Spanish-speaking community (Community 1) suggests that language is a cultural determinant that biases the formation of the depicted structure of TTs coverage. To confirm this hypothesis we have divided the countries in our TT-2014 datasets into three groups based on their official languages: ”Spanish-speaking” countries, ”English-speaking” countries and ”Others”. Figure 88 shows the distribution of the leader-follower links weights across countries within each group in the form of a boxplot. The results confirm that leader-follower relationships are stronger between countries with a common language. Furthermore, the results suggest that Spanish creates stronger relationships than English, explaining the presence of a community that includes all Spanish-speaking countries in TT-2014.

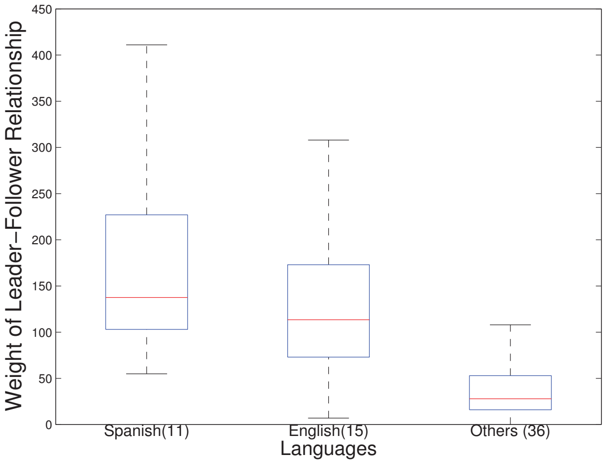


Figure 88: Distribution of the weight of leader-follower relationships for countries grouped by language (The number in brackets refers to the number of members within the group)

3.2.5 Mass Media and Twitter Interaction in News Reporting

TTs capture events of national interest that may be also reported by national mass media. In this section, we present a methodology that leverages Google News service [GNS] in order to identify the overlapping news between national mass media and TTs for a given country. We refer to these TTs as External TTs as opposite to those news that appear exclusively as TTs in Twitter, and thus are not reported by mass media that we refer to as Internal TTs. We have applied our methodology to the TTs of 4 countries (US, GB, CA and ES) collected over a period of one month. We leverage the obtained results to understand the influence of mass media in news reported as TTs in Twitter and the extent to which Twitter is used as an independent channel for the propagation of exclusive news not reported by mass media. Moreover, we provide quantitative evidences of which venue, Twitter or traditional mass media, reports earlier overlapping news appearing in both.

3.2.5.1 Methodology

A TT appearing in country C at a given date d is External if an associated piece of news appears in other traditional mass media (e.g., a newspaper) of the same country in a small time window around d (e.g., ± 2 days). Otherwise, it can be considered Internal and not newsworthy for traditional media.

Based on this definition, we leverage the Google News service [GNS], which provides all the news reported by the most important media of a country in a given time window, in order to identify whether the analyzed TT appeared also in other media and thus it should be marked as External.

For a given TT our methodology is divided into a pre-processing phase to translate the TT into an appropriate format to query the Google News service, and a search phase in which the Google News service seeks for news including the words forming our TT. Next we describe in detail the operations done during each phase.

- The pre-processing phase is divided into two steps. First, (when required) we transform the TT in a set of meaningful words. For instance a TT “#BarackObamaInNewYork” would be transformed into “Barack Obama in New York” (the # is removed and the words forming the TT are properly separated). Second, for every word obtained in the previous step, we check if it is included in the list of the Top 1000 most frequent words in the language of the country. If all the words forming a TT are among the Top 1000 most frequent words of the language, we filter out that TT and do not consider it in our analysis.

- In the search phase we query the Google News service with the following information: (i) we use as keyword(s) the set of words obtained from the pre-processing phase; (ii) the specific media outlet to make the search; (iii) a time window to perform the search. Furthermore, we force the Google News service to search the keyword(s) in both the headline and the body of the articles. The Google News service returns all those news that include all the keyword(s) indicated in the query in either the headline or the body. It also provides the date of appearance of each returned piece of news.
- Therefore, if as result of the search phase the Google News service returns at least one piece of news, we classify the associated TT as External. We mark the TT as Internal otherwise. Moreover, for External TTs with several associated news, we consider that the piece of news appeared in external media on the earliest date among the associated news. We can analyze if news usually appear earlier in Twitter or in mass media comparing the appearance dates of External TTs and their associated news.

We have manually evaluated the accuracy of the described methodology using a random set of 1000 TTs from Spain. Specifically, three independent panelists subjectively classified each TT as Internal or External. To the best of our knowledge, there is not an automated technique able to perform accuracy tests for this type of contextual classification exercises. Then, we rely on human subjective validation that has also been used in previous works [CGC+13, LPN+11, ZSF+11]. The obtained results indicate that our methodology classified correctly 96% of the TTs and thus we can conclude that it provides accurate results.

We have applied the described methodology to the TTs of four countries (US, ES, CA and GB) collected over a period of time of 1 month between Apr-14-2014 and May-12-2014. In particular, US, ES, CA and GB presented 5297, 2598, 2385 and 4598 TTs during this period, respectively. Furthermore, we have used a time window of ± 2 days around the date of the TT appearance to search for associated news in the online version of three main newspapers of each country. A 5-days time window seems to be a conservative period around a breaking new such as a TT to be reported by traditional mass media. We have repeated the experiments for a time window of 7 days obtaining similar results.

3.2.5.2 Influence of Mass Media News in Twitter TTs

Figure 89 depicts the percentage of External TTs for each one of the analyzed countries. The results demonstrate that Twitter is not isolated from mass media, since slightly more than half of the TTs are associated to news also reported in mass media. Despite this important interaction between Twitter and mass media, still roughly between 40-46% of the TTs are associated to events or news only reported in Twitter. Our data shows that hashtaging is a clear differentiating factor between Internal and External TTs. Figure 89 presents the percentage of Internal and External TTs that are hashtags across the four studied countries. We observe that roughly 60-80% Internal TTs are hashtags whereas these values shrink to 10-25% for External TTs. This suggests that in order to become TT, without the back up of news reported by mass media, a communication mechanism is required. The hashtaging functionality seems to be such mechanism in Twitter, allowing both the collective discussion about the topic and the semiotic creation of a symbol to refer to it [LGR+12].

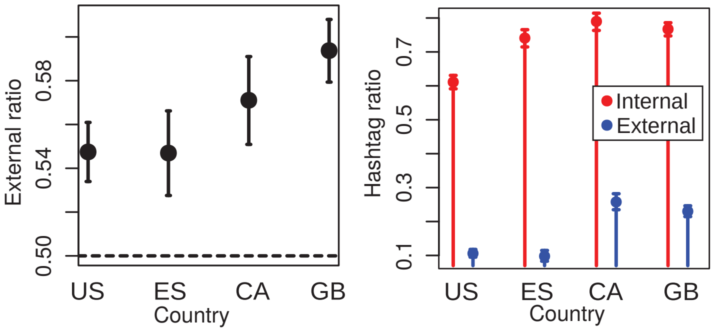


Figure 89: (Left) Ratio of external TTs; (Right) Ratio of external and internal TTs that are hashtags. Considered countries US, ES, CA and GB

3.2.5.3 *Twitter as an Independent Media for International News Coverage*

We have analyzed the TTs shared by each pair of the four considered countries. Note that if a TT is Internal in both countries, then it was shared internally within Twitter and it was not reported by traditional media in either country. Moreover, if a shared TT is External in any of the two countries, the sharing process may have occurred through either mass media or Twitter. Then, to be conservative, we consider that the TT was shared through external media in this case. Based on these considerations, Figure 90 shows the conditional probability that a TT is shared internally within Twitter or externally through other media. We confirm that, for the four analyzed countries, a TT is more likely to be shared internally. Indeed, the probability of being shared internally is 50-100% higher than externally. In conclusion, the flow of TTs between countries is mainly formed by internal news generated in Twitter rather than news of interest to traditional mass media. This confirms the role of Twitter as an alternative communication venue to mass media.

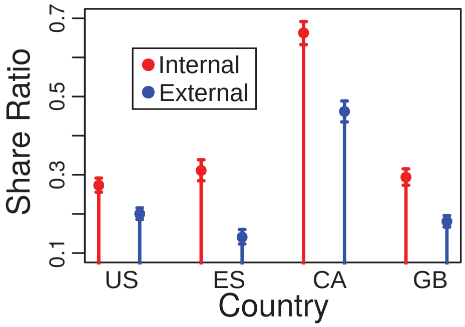


Figure 90: Conditional probability of shared TTs being internal for US, ES, CA and GB

3.2.5.4 *Recency in News Reporting: Twitter vs. Mass Media*

To conclude our analysis, in this subsection, we compare the surging dates of External TTs and their associated news in mass media to check in which communication venue news are (typically) reported earlier. Specifically, we have computed the difference in number of days between the appearance date of every external TT and its associated piece of news in online newspapers in the four considered countries. Note that the difference ranges between ± 2 days from the appearance date of the TT since this is the window time that we have configured to query the Google News service. Figure 91 shows the percentage of news that surged two days before, one day before, the same day, one day after and two days after in traditional media than in Twitter. Roughly 70-75% news are reported earlier in Twitter than in the online edition of main newspapers whereas less than 10% of the news appeared earlier in those newspapers. The rest appeared the same day in both venues. These results highlight the role of Twitter as a main venue for breaking news reporting.

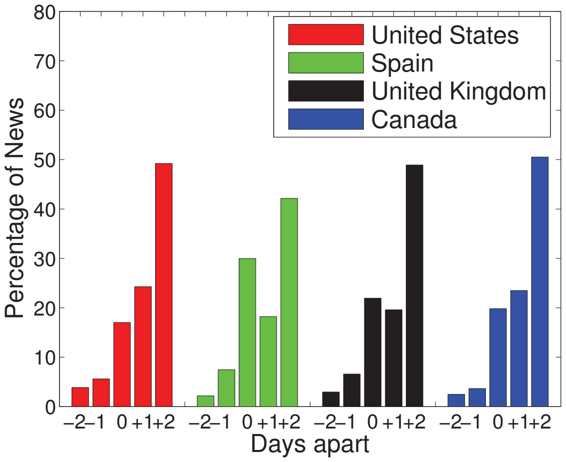


Figure 91: Percentage of news appearing 2 days before, 1 day before, the same day, 1 day after or 2 days after in principal newspapers than in Twitter for US, ES, CA and GB

3.2.6 Related Work

The analysis of propagation of different pieces of information in social media has received the attention of the research community in the last years. Huffaker et al. [HTS11] analyze the diffusion process from the perspective of the importance of the role of groups in virtual worlds (e.g., Second Life). Moreover, the impact of social networks on users' behaviour and thus, indirectly, in the information dissemination is analyzed in [BRM+12]. Other studies have specifically focused in the geographical propagation of information in social media including the flow of music [LC12] or the evolution of information pathways in the online media space (e.g., blog sites) [GLS13]. These studies analyze the spread of different types of pieces of information in various social venues not covered by our work and thus their results are complementary to ours.

Of more interest to this research, other works have focused on the analysis of information propagation in Twitter. Scellato et al. [SMM+11] study the social cascades of Youtube links in Twitter. Cha et al. [CHB+10] use the spread of popular news topics to investigate influence in Twitter. Kamath et al. [KCL+13] have analyzed the geographical diffusion of hashtags in Twitter. The paper concludes that hashtags are mostly a local phenomenon. We have also observed a high locality among TTs. Moreover, Wilkinson et al. [WT12] analyze the external socio-economic factors that influence the propagation of popular topics in Twitter (among english-speaking countries). Saez-Trumper et al. [STCL13] have demonstrated a strong correlation of the coverage distribution of different pieces of information with geographical regions. They also conclude that sources from a given geographical region tend to share the same stories. Romero et al. [RMK11] have classified Twitter hashtags in 8 different categories and have analyzed how these hashtags spread over the Twitter users interactions' network finding significant variation depending on the category. Finally, the work by An et al. [ACG+11] studies the media landscape in Twitter. The authors conclude that traditional media has an important presence in the media landscape in Twitter. This result is aligned with our observation of a high overlapping between TTs and news reported by mass media. To the best of the authors' knowledge, Ferrara et al. [FVM13] performed the only work addressing the geographical coverage of TTs in Twitter. However, they focus on the internal propagation of TTs among cities in US and do not analyze the socio economic biases of the resulting structure. In general, these studies target different type of information (e.g., Youtube links or hashtags), different geographical scope (e.g., english-speaking countries or cities in US) or focus exclusively on Twitter whereas in our research we analyze TTs as a piece of information with a worldwide scope and also analyze the interaction of Twitter and traditional mass media. In this context, Kwak et al. [KLP+10] analyze the overlapping between TTs and mass media news in 2009. First, the authors confirm the condition of breaking news of TTs. Furthermore, they compare news coverage by World Wide Twitter TTs with Hot Topics from Google Trends and headlines of CNN news concluding that CNN was ahead in reporting. Our results reveal that this situation has been reversed in the last years, and now Twitter seems to be ahead mass media in news reporting.

Finally, the research community has dedicated an important amount of effort to understand the structure that drives the international coverage of traditional news over mass media as well as the determinants that influence the news coverage [EKP08, GJW10, W00, W93]. The conducted research conclude that there is an established structure in the propagation of news that (typically) flows from a set of core countries towards other countries located in the semi-periphery and periphery. Furthermore, factors such as the GDP or the trade volume of a country determine its role in the news coverage structure.

The goal of our research is of similar nature, but we focus on the international coverage of news in social media channels using TTs as reference.

3.2.7 Conclusion

We showed how news in social media manifest through Local TTs in Twitter, analyzing two alternative large-scale datasets of TTs in different countries. We validated our analysis of the leader-follower relationships between countries testing the hypothesis of priority processes in queue systems [OB05, VOD+06], finding a power-law distribution of delay times between the appearance of TTs. This finding conveys knowledge about the dynamics of how TTs travel across countries, in an analogous manner as how power-law degree distributions reveal dynamics of preferential attachment or edge copying mechanisms [M04]. Applying the statistical physics of priority processes has potential applications to the analysis of communication dynamics in other online communities, from dialogues to collective reactions.

Leader-follower relationships among countries reveal patterns of heterogeneity in both influence and attention, allowing us to test hypotheses inspired in mass media about the role of economic and social factors in news coverage [O65, HC08, I96]. We found that news in social media follow the gradient of wealth from rich to poor countries, and that TT across countries are closely related to migrations across them. Our combination of data about TT with Hofstede's quantification of culture [LBH80] contributes to the wider scientific field of online ethnography [K02] and resonates with works about the online manifestation of cultural traits [GQJ13, WSS14] and cultural affinity [GT13].

Our addition of cultural factors adds a new dimension to the analysis of social media usage across nations. The modular structure of the TTs network is clustered around communities with high internal cultural similarity, and longer cultural distances across communities. This inspires future possible validation of clustering and community detection techniques at the international level, in which cultural distances can be taken as ground truth to evaluate community detection algorithms.

To analyze the role of mass media in TTs, we designed a method to match TTs to news in mass media close to the TT appearance. Using this method, we found that internal TTs are much more likely to manifest around a hashtag, which serves as a symbol to centralize communication in the absence of important mass media channels. This method also allowed us to statistically control for mass media in how TTs are shared across countries, revealing that external TTs are less likely to cross country borders in Twitter than those TTs that were not considered newsworthy by the mass media. Further applications of this tool have the potential to enhance the analysis of dynamic collective response patterns in Twitter [LGR+12], allowing the measurement of reach and social interaction around news channels.

Our work shows how news in social media can break international borders through Twitter TTs, revealing that Twitter is used as an alternative communication channel with respect to mass media.

On the other hand, we found significant biases with respect to economic, demographic, and cultural factors. This portrays Twitter as a mixed and multipurpose community, in which news can flow without constraints, but also in which mass media have a strong influence that replicates the same biases as previously found in traditional media research.

4. SUMMARY AND CONCLUSIONS

In the scope of WP3, active crawling and passive measurement tools have been developed for online social networks such as Facebook, Twitter, and Google+, as well as content portals and content distribution networks such as YouTube, BitTorrent, and BTLive, which were used for data collection. Specifically, a number of datasets have been collected using these tools, which were exploited in WP3, as well as in WP4 and WP5.

Based on the analysis of these datasets we were able to draw a number of conclusions related to modelling social-content interdependencies. To this end, Section 2.1 presented a methodology to analyze interest similarity among FB users and studied the homophily of such similarity using different demographic metrics. Moreover, Section 2.2 presented a model to predict the location of a user based on the public available information of that user and the information of their friends in Facebook. In addition, a novel methodology to measure the location exposure level for Facebook end-users was proposed. Section 2.3 demonstrated that professional FB players present different publishing strategies in major OSNs according to the sector they belong to. We characterized the strategy of some sectors and evaluated their success.

Furthermore, Section 2.4 analyzed the characteristics of BTLive. The work showed that the P2P effect takes over with an increasing number of peers in the swarm. However, there exists great potential to reduce the load at the source by more aggressively dropping connections. It was also shown that BTLive is able to maintain very small startup and streaming delays. However, those delays come at a high cost in terms of traffic overhead. Section 2.5 identified different predictive features for videos shared over Online Social Networks. It turned out that, e.g., the video category is an efficient criterion for mobile prefetching, as well the related video section of YouTube, while the number of likes and comments has turned out to be less predictive. At the same time, Section 2.6 showed that video platforms present important deficiencies in the detection of fake views. This may seriously impact the capacity of predicting the right location where a content is going to be consumed and thus leading the existing caching and prefetching algorithms to make wrong decisions. A detailed analysis of the detection algorithm of YouTube (the most important online video portal) has revealed that the most important aspect considered is the behaviour of the IP address whereas our results demonstrate that other relevant parameters such as the duration of the view are not considered.

Furthermore, Section 2.7 identified that D2D content delivery is only effective under certain circumstances. It is important to investigate if these circumstances are given in the real world. Thus, we described a model for the evaluation of position and request traces of a mobile network operator, by *simulating* D2D content exchange on the collected user behaviour. Finally, Section 2.8 showed that the model based on learning a feature vector for each user and each content item and where the prediction consists of a dot product of the feature vector associated with the user and content item, outperforms the model in which the prediction is based on making an average within a set of close users and content items.

We have also used the datasets and derived models to predict certain characteristics. To this end, the comparative analysis of the social cascades in Google+ and Twitter reported in Section 3.1 revealed that in any major OSN only a very small percentage of the information is propagated. Moreover, the specific functionality of each OSN defines the spatio-temporal properties of its social cascades. The sequential manner of showing the tweets in Twitter leads to social cascades of small size and length and of short duration of times. Instead, the selective algorithm used by Google+ to show information to the users makes that both the size and the duration of the social cascades is larger in Google+ than in Twitter. In addition, we have presented a pioneer study in the area of social cascades to study the social cascades in the form of forests with multiple roots for a specific content. Our analysis of the overlapping of social cascades and the social graph indicates that around 70% of the information propagated in Google+ does not use the social graph and instead it is propagated through other venues such as communities or hashtags. Finally, in Section 3.2 the analysis of the macroscopic information flow in social networks through Trending Topics have revealed the presence of clear communities of countries that tend to share a higher volume of information. These communities seem to be driven by socio-economic factors. Moreover, the comparison between Twitter and traditional mass media indicates that Twitter can be considered as an independent channel of information. Besides, for those pieces of information that appear both in Twitter and mass media, around 70% of them appear earlier in Twitter.

REFERENCES

[AA03] L. Adamic and E. Adar, Friends and neighbors on the Web, Social Networks, vol. 25, no. 3, pp. 211–230, Jul 2003.

[ABP14] J. Alstott, E. Bullmore, and D. Plenz. Powerlaw: a python package for analysis of heavy-tailed distributions. PloS one, 9(1):e85777, 2014.

[ACG+11] J. An, M. Cha, P. Gummadi, and J. Crowcroft. Media landscape in twitter: A world of new conventions and political diversity. In ICWSM, 2011.

[ADF+07] A. Arenas, J. Duch, A. Fernandez, and S. Gómez. Size reduction of complex networks preserving modularity. New Journal of Physics, 9(6):176–176, June 2007.

[AGL09] E. Alessandria, M. Gallo, E. Leonardi, M. Mellia, and M. Meo, P2P-TV Systems under Adverse Network Conditions: A Measurement Study, in IEEE INFOCOM, 2009.

[AK10] S. Abrol and L. Khan, Tweethood: Agglomerative clustering on fuzzy k-closest friends with variable depth for location mining, in SocialCom, 2010, pp. 153–160.

[AKT12] S. Abrol, L. Khan, and B. Thuraisingham, Tweecalization: Efficient and intelligent location mining in twitter using semi-supervised learning, in CollaborateCom, 2012, pp. 514–523.

[Anw14] J. Anwer, Now, watch youtube even when you are offline, September 2014, http://indiatoday.intoday.in/technology/story/now-watch-youtube-even-when-you-are-ffline/1/382937.html.

[AQC+14] J. An, D. Quercia, M. Cha, K. Gummadi, and J. Crowcroft. Sharing political news: the balancing act of intimacy and socialization in selective exposure. EPJ Data Science, 3(1):1–21, 2014.

[AQC14] J. An, D. Quercia, and J. Crowcroft. Partisan sharing: facebook evidence and societal consequences. In Proceedings of the second edition of the ACM conference on Online social networks, pages 13–24. ACM, 2014.

[ASA14] Annual Shorty Awards Winners. http://shortyawards.com/.

[AW] Google adwords. https://adwords.google.com. [27] [EOS+05] B. Edelman, M. Ostrovsky, M. Schwarz, T. D. Fudenberg, L. Kaplow, R. Lee, P. Milgrom, M. Niederle, and A. Pakes, Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords, American Economic Review, vol. 97, 2005.

[AW14] Fraud Alert: Millions of Video Views Faked in Sophisticated New Bot Scam. http://www.adweek.com/news/technology/fraud-alert-millions-video-views-faked-sophisticated-new-bot-scam-156883, 2014.

[B13] J. Brown, The Click Fraud Report Infographic: First Half 2013. http://www.adometry.com/ blog/click-fraud-report-infographic/, 2013.

[BBR+12] L. Backstrom, P. Boldi, M. Rosa, J. Ugander, and S. Vigna. Four degrees of separation. In ACM WebSci, 2012.

[BMB11] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, The socialbot network: When bots socialize for fame and money, in Proceedings of the 27th Annual Computer Security Applications Conference, ACSAC ’11, (New York, NY, USA), pp. 93–102, ACM, 2011.

[BMH+11] M. S. Bernstein, A. Monroy-Hernández, D. Harry, P. Andŕe, K. Panovich, and G. G. Vargas. 4chan and/b: An analysis of anonymity and ephemerality in a large online community. In ICWSM, 2011.

[BR87] J. J. Brown and P. H. Reingen. Social ties and word-of-mouth referral behavior. Journal of Consumer Research, pages 350–362, 1987.

[BRG+11] J. Borge-Holthoefer, A. Rivero, I. Garcıa, E. Cauhe, A. Ferrer, D. Ferrer, D. Francos, D. Iñiguez, M. Pilar Ṕerez, G. Ruiz, et al. Structural and dynamical patterns on online social networks: the spanish may 15th movement as a case study. PloS one, 6(8):e23883, 2011.

[BRM+12] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic. The role of social networks in information diffusion. In WWW. ACM, 2012.

[BSB09] L. Bilge, T. Strufe, D. Balzarotti, and E. Kirda, All your contacts are belong to us: Automated identity theft attacks on social networks, in Proceedings of the 18th International Conference on World Wide Web, WWW ’09, (New York, NY, USA), pp. 551–560, ACM, 2009.

[BSM10] L. Backstrom, E. Sun, and C. Marlow, Find me if you can: Improving geographical prediction with social and spatial proximity, in WWW, 2010, pp. 61–70.

[BT10] M. Barlow and D. Thomas, In The Executive’s Guide to Enterprise Social Media Strategy: How Social Networks Are Radically Transforming Your Business. John Wiley and Sons, 2010.

[CAA+09] M. Cha, A. Mislove, B. Adams, and K. P Gummadi. Characterizing social cascades in flickr. In ACM WOSN, 2008.

[CAG09] M. Cha, A. Mislove, and K. P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In WWW, 2009.

[CAGW] Ibiblio irc logs: Ussr coup attempt and gulf war. http://www.ibiblio.org/pub/academic/communications/logs/.

[CBW] The cuttlefish network workbench. http://cuttlefish.sourceforge.net/.

[CCL12] Z. Cheng, J. Caverlee, and K. Lee, You are where you tweet: A content-based approach to geo-locating twitter users, in CIKM, 2012, pp. 759–768.

[CCR+03] B. Chun, D. Culler, T. Roscoe, A. Bavier, L. Peterson, M. Wawrzoniak, and M. Bowman, Planetlab: An overlay testbed for broad-coverage services, SIGCOMM Comput. Commun. Rev., vol. 33, pp. 3–12, July 2003.

[CDJ08] X. Cheng, C. Dale, and J. Liu, Statistics and social network of youtube videos, in in Proc. of IEEE IWQoS, 2008.

[CDK+03] M. Castro, P. Druschel, A.M. Kermarrec, et al., SplitStream: High-Bandwidth Multicast in Cooperative Environments, in ACM SIGOPS, 2003.

[CGC+13] J. Carrascosa, R. Gonzalez, R. Cuevas, and A. Azcorra. Are Trending Topics Useful for Marketing? Visibility of Trending Topics vs Traditional Advertisement. In COSN, 2013.

[CHB+10] M. Cha, H. Haddadi, F. Benevenuto, and P. Gummadi. Measuring user influence in twitter: The million follower fallacy. ICWSM, 2010.

[CHC13] W. Chanthaweethip, X. Han, N. Crespi, Y. Chen, R. Farahbakhsh, and A. Cuevas, Current city prediction for coarse location based applications on facebook, in GLOBECOM, 2013, pp. 3188–3193.

[CHP+14] C. Castillo, M. El-Haddad, J. Pfeffer, and M. Stempeck. Characterizing the life cycle of online news stories using social media reactions. In Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work &#38; Social Computing, CSCW ’14, pages 211–223, 2014.

[Cis13] Cisco, Cisco Visual Networking Index: Forecast and Methodology, 2012–2017, Tech. Rep., 2013.

[Cis14] Cisco, Cisco Visual Networking Index: Forecast and Methodology, 2013–2018, Tech. Rep., 2014.

[CJB08] Z. Chen, C. Ji, and P. Barford, Spatial-temporal characteristics of internet malicious sources, in INFOCOM 2008. The 27th Conference on Computer Communications. IEEE, April 2008.

[CKG+14] R. Cuevas, M. Kryczka, R. González, A. Cuevas, and A. Azcorra, Torrentguard: Stopping scam and malware distribution in the bittorrent ecosystem, Comput. Netw., vol. 59, pp. 77–90, Feb. 2014.

[CKM11] S. Chandra, L. Khan, and F. Muhaya, Estimating twitter user location using social interactions–a content based approach, in SocialCom, 2011, pp. 838–843.

[CL09] X. Cheng and J. Liu, Nettube: Exploring social networks for peer-to-peer short video sharing, in INFOCOM 2009, IEEE, April 2009, pp. 1152–1160.

[CMC+13] R. Cuevas, M. Kryczka, A. Cuevas, S. Kaune, C. Guerrero, and R. Rejaie, Unveiling the incentives for content publishing in popular bittorrent portals, IEEE/ACM Trans. Netw., vol. 21, pp. 1421–1435, Oct. 2013.

[CMM14] M. D. Choudhury, A. Monroy-Hernandez, and G. Mark. Narco emotions: affect and desensitization in social media during the mexican drug war. In Proceedings of the 32nd annual ACM conference on Human factors in computing systems, pages 3563–3572. ACM, 2014.

[Coh03] B. Cohen, Incentives Build Robustness in BitTorrent, in Workshop on Economics of Peer-to-Peer systems (P2PECON), 2003.

[Coh12] B. Cohen, How to Do Live P2P Video Streaming, 2012, Keynote Speech at IEEE International Conference on Peer-to-Peer Computing. [Online]. Available: http://www.p2p12.org/program/keynote-speakers

[Coh13] B. Cohen, Peer-to-Peer Live Streaming, Patent 20 130 066 969, March, 2013. [Online]. Available: http://www.freepatentsonline.com/y2013/ 0066969.html

[CPA] CPA Detective. http://cpadetective.com/advertisers.html.

[CRN08] A. Clauset, C. R. Shalizi, and M. EJ Newman. Power-law distributions in empirical data. SIAM review, 51(4):661–703, 2009.

[CSB87] T. Chang, P. J. Shoemaker, and N. Brendlinger. Determinants of international news coverage in the us media. Communication Research, 14(4):396–414, 1987.

[CTH10] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, Bridging the gap between physical location and online social networks, in UbiComp, 2010, pp. 119–128.

[CWY12] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In ACM SIGKDD, 2009.

[CZC14] L. Chen, Y. Zhou, and D. M. Chiu, Fake view analytics in online video services, in Proceedings of Network and Operating System Support on Digital Audio and Video Workshop, NOSSDAV ’14, (New York, NY, USA), pp. 1:1–1:6, ACM, 2013.

[D3.1] EU eCOUSIN: Public Deliverable D3.1 - Measurement, Modelling & Prediction of Social-Content Interdependencies, October 2013.

[D3.2] Initial Release of Measurement and Prediction Software, eCOUSIN Deliverable D3.2, April 2014.

[D4.2] Final Report and Initial Software Release of the Design Extensions and Preliminary Implementation, eCOUSIN Deliverable D4.2, April 2014.

[DGZ12] V. Dave, S. Guha, and Y. Zhang, Measuring and fingerprinting click-spam in ad networks, SIGCOMM Comput. Commun. Rev., vol. 42, pp. 175–186, Aug. 2012.

[DGZ13] V. Dave, S. Guha, and Y. Zhang, Viceroi: Catching click-spam in search ad networks, in Proceedings of the 2013 ACM SIGSAC Conference on Computer; Communications Security, CCS '13, (New York, NY, USA), pp. 765–776, ACM, 2013.

[DK06] M. Duckham and L. Kulik, Location privacy and location-aware computing, Dynamic & mobile GIS: investigating change in space and time, vol. 3, pp. 35–51, 2006.

[DLL+00] C. Diot, B. N. Levine, B. Lyles et al., Deployment Issues for the IP Multicast Service and Architecture, IEEE Network, vol. 14, no. 1, pp. 78–88, 2000.

[DM13] M. W. DiStaso and T. McCorkindale, A benchmark analysis of the strategic use of social media for fortunes most admired u.s. companies on facebook, twitter, and youtube. Public Relations Journal, 7(1), 1-33 (2013).

[Dre13] T. Dreier, BitTorrent Recruits Testers for Live Video Platform, Oct. 2013, Interview with Tim Leehane, BitTorrent Inc., Streaming Media East Conference. [Online]. Available: http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/ BitTorrent-Recruits-Testers-for-Live-Video-Platform-92757.aspx

[EKP08] N. T. Ekeanyanwu, Y. KalyangoJnr, and A. S. Peters. Global News Flow Debate in the Era of Social Media Networks: Is the US Me Still the World's News Leader? European Scientific Journal, 8(3).

[eM13] eMarketer, Online Video Advertising Moves Front and Center. http://www.emarketer.com/Article/Online-Video-Advertising-Moves-Front-Center/1009886, May 2013.

[eM14] eMarketer, As Barriers Tumble, Video Marketing Adoption Grows. http://www.emarketer.com/Article/Barriers-Tumble-Video-Marketing-Adoption-Grows/1010374, 2014.

[eM213] Advertisers to Spend $5.60 Billion on YouTube in 2013 Worldwide. http://www.emarketer.com/Article/Advertisers-Spend-560-Billion-on-YouTube-2013-Worldwide/1010446, 2013.

[Eri12] Ericsson ConsumerLab, TV and Video -An analysis of evolving consumer habits. Tech. Rep., 2012.

[Eri13] Ericsson, TV and Media -Identifying the Needs of Tomorrow's Video Consumers, Tech. Rep., 2013.

[EVA10] L. Evans, In Social Media Marketing: Strategies for Engaging in Facebook, Twitter & Other Social Media. Pearson Education, 2010.

[FFL12] L. Fang, A. Fabrikant, and K. LeFevre. Look who i found: understanding the effects of sharing curated friend groups. In ACM WebSci, 2012.

[FHC13] R. Farahbakhsh, X. Han, A. Cuevas, and N. Crespi, Analysis of publicly disclosed information in facebook profiles, in ASONAM, 2013, pp. 699–705.

[FLIX] http://www.cs.sfu.ca/~sja25/personal/datasets/

[FMM11] A. Finamore, M. Mellia, M. M. Munaf`o, R. Torres, and S. G. Rao, Youtube everywhere: Impact of device and infrastructure synergies on user experience, in Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, ser. IMC '11. NY, USA: ACM, 2011, pp. 345–360.

[FVM13] E. Ferrara, O. Varol, F. Menczer, and A. Flammini, Traveling trends: Social butterflies or frequent fliers? COSN '13, ACM (2013).

[G07] M. M. Geipel. Self-organization applied to dynamic network layout. International Journal of Modern Physics C, 18(10):1537–1549, 2007.

[G12] D. Gayle, YouTube cancels billions of music industry video views after finding they were fake or ’dead’. http://www.dailymail.co.uk/sciencetech/article-2254181/ YouTube-wipes-billions-video-views-finding-faked-music-industry.html, 2012.

[G14] S. Gutelle, The Average YouTube CPM Is $7.60, But Making Money Isn’t Easy. http://www.tubefilter.com/2014/ 02/03/youtube-average-cpm-advertising-rate/, 2014.

[G64] B. S. Greenberg. Person-to-person communication in the diffusion of news events. Journalism & Mass Communication Quarterly, 41(4):489–494, 1964.

[GCR+13] R.Gonzalez, R. Cuevas, R. Rejaie, R. Motamedi, A. Cuevas. Google+ or Google-?: Dissecting the Evolution of the New OSN in its First Year. In WWW, 2013.

[GGS+12] A. Garas, D. Garcia, M. Skowron, and F. Schweitzer. Emotional persistence in online chatting communities. Scientific Reports, 2, 2012.

[GGS14] M. Gomez-Rodriguez, K. P. Gummadi, and B. Scholkopf. Quantifying information overload in social media and its impact on social contagions. 2014.

[GHM+12] N. Z. Gong, W. Xu, L. Huang, P. Mittal, E. Stefanov, V. Sekar, and D. Song. Evolution of attribute-augmented social networks: Measurements, modeling, and implications using google+. In ACM IMC, 2012.

[GJA09] S. Gómez, P. Jensen, and A. Arenas. Analysis of community structure in networks of correlated data. Physical Review E, 80(1):016114, July 2009.

[GJW10] G. Golan, T. Johnson, and W. Wanta. Determinants of international news coverage. International media communication in a global age, pages 125–144, 2010.

[GKB11] M. Gjoka, M. Kurant, C. Butts, and A. Markopoulou, Practical recommendations on crawling online social networks, IEEE JSAC, vol. 29, no. 9, pp. 1872–1892, 2011.

[GKS+13] C. Gross, F. Kaup, D. Stingl, B. Richerzhagen, D. Hausheer, and R. Steinmetz, Enersim: An energy consumption model for large-scale overlay simulators. in LCN, 2013, pp. 252–255.

[GLK10] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In ACM SIGKDD, 2010.

[GLS13] M. G. Rodriguez, J. Leskovec, and B. Scholkopf. Structure and dynamics of information pathways in online media. In WSDM, 2013.

[GLx13] G. Gao, R. Li, W. Xiao, and Z. Xu, Measurement Study on P2P Streaming Systems, Springer Journal of Supercomputing, vol. 66, no. 3, pp. 1656–1686, Jun. 2013.

[GNS] Google News Service. https://news.google.com/.

[Goo] Views report. https://support.google.com/youtube/answer/1714329.

[Goo14] Frozen view count. https://support.google.com/youtube/troubleshooter/2991876.

[GooD] Reference Guide: Data API Protocol. https://developers.google.com/youtube/2.0/reference.

[GooS14] Keeping YouTube Views Authentic. http://googleonlinesecurity.blogspot.co.uk/ 2014/02/keeping-youtube-views- authentic.html, 2014.

[GQJ13] R. Garcia-Gavilanes, D. Quercia, and A. Jaimes. Cultural dimensions in twitter: Time, individualism and power. AAAI ICWSM, 2013.

[GR65] J. Galtung and M. H. Ruge. The structure of foreign news the presentation of the congo, cuba and cyprus crises in four norwegian newspapers. Journal of peace research, 2(1):64–90, 1965.

[GT13] D. Garcia and D. Tanase. Measuring cultural dynamics through the eurovision song contest. Advances in Complex Systems, 16(08), 2013.

[GWR+14] D. Garcia, I. Weber, and R. Garimella. Gender asymmetries in reality and fiction: The bechdel test of social media. In International AAAI Conference on Weblogs and Social Media, pages 131–140, 2014.

[H12] C. Hoffberger, YouTube strips Universal and Sony of 2 billion fake views. http://www.dailydot.com/news/youtube-universal-sony-fake-views-black-hat/, 2012.

[H13] Bernardo A Huberman. Social computing and the attention economy. Journal of Statistical Physics, 151(1-2):329–339, 2013.

[HAJ12] H. Hu, G. Ahn, and J. Jorgensen. Enabling collaborative data sharing in google+. In IEEE GLOBECOM, 2012.

[HC08] E. S. Herman and N. Chomsky. Manufacturing consent: The political economy of the mass media. Random House, 2008.

[HLL+13] X. Hei, C. Liang, J. Liang, Y. Liu, and K. Ross, A Measurement Study of a Large-Scale P2P IPTV System, IEEE Transactions on Multimedia, vol. 9, no. 8, pp. 1672–1687, 2007.

[HRW08] B. Huberman, D. Romero, and F. Wu. Social networks that matter: Twitter under the microscope. Available at SSRN 1313405, 2008.

[HTF01] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, 2001.

[HTS11] D. A. Huffaker, C. Teng, M. P. Simmons, L. Gong, and L. A. Adamic. Group membership and diffusion in virtual worlds. In IEEE Third international conference on social computing (socialcom). IEEE, 2011.

[I96] K. Ish. Is the us over-reported in the japanese press? factors accounting for international news in the asahi. International Communication Gazette, 57(2):135–144, 1996.

[IAB1] IAB internet advertising revenue report, 2013 full year results. http://www.iab.net/media/file/IAB_Internet_Advertising_Revenue_Report_FY_2013.pdf.

[IAB2] IAB AntiFraud Working Group. http://www.iab.net/about_the_iab/recent_press_releases/press_release_archive/press_release/pr-091614.

[IMP] Improvely. https://www.improvely.com.

[IVR13] Y. Ikawa, M. Vukovic, J. Rogstadius, and A. Murakami, Location-based insights from the social web, in WWW Companion, 2013, pp. 1013–1016.

[J13] D. de Jager, Display Advertising Fraud is a Sell-Side Problem. http://www.spider.io/blog/2013/04/display-advertising-fraud-is-a-sell-side-problem/, 2013.

[JCG+14] J. Carrascosa, R. Cuevas, R. Gonzalez, A. Azcorra, and D. Garcia. Quantifying geographic news coverage and its underlying biases in social media through Trending Topics. Technical report available at: http://www.it.uc3m.es/~rcuevas/techreports/TT_ propagation_TR2014.pdf, Universidad Carlos III de Madrid, 2014.

[K02] R. V. Kozinets. The field behind the screen: using netnography for marketing research in online communities. Journal of marketing research, 39(1):61–72, 2002.

[K14] A. Kantrowitz, Ad-Fraud Operation Fools Detection Companies, Nets Millions. http://adage.com/article/digital/ad-fraud-operation-fools-detection-companies-nets-millions/293929/, 2014.

[KA14] H. Kwak and J. An. Understanding news geography and major determinants of global news coverage of disasters. arXiv preprint arXiv:1410.3710, 2014.

[KBH+12] S. Kairam, M. Brzozowski, D. Huffaker, and E. Chi. Talking in circles: selective sharing in google+. In SIGCHI, 2012.

[KCL+13] K. Y. Kamath, J. Caverlee, K. Lee, and Z. Cheng. Spatio-temporal dynamics of online memes: a study of geo-tagged tweets. In www, 2013.

[KKT03] D. Kempe, J. Kleinberg, and E ́. Tardos. Maximizing the spread of influence through a social network. In ACM SIGKDD, 2003.

[KLP+10] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In WWW, 2010.

[KN12] S. S. Krishnan and R. K. Sitaraman, Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs, in Proceedings of the 2012 ACM Conference on Internet Measurement Conference, ser. IMC ’12. New York, NY, USA: ACM, 2012, pp. 211–224. http://doi.acm.org/10.1145/2398776.2398799

[KRH07] A. Karasaridis, B. Rexroad, and D. Hoeflin, Wide-scale botnet detection and characterization, in Proceedings of the First Conference on First Workshop on Hot Topics in Understanding Botnets, HotBots’07, (Berkeley, CA, USA), pp. 7–7, USENIX Association, 2007.

[KS13] S. S. Krishnan and R. K. Sitaraman, Understanding the effectiveness of video ads: A measurement study, in Proceedings of the 2013 Conference on Internet Measurement Conference, IMC ’13, (New York, NY, USA), pp. 149–162, ACM, 2013.

[KZG13] D. K. Krishnappa, M. Zink, and C. Griwodz, What should you cache?: A global analysis on youtube related video caching, in Proceeding of the 23rd ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, ser. NOSSDAV ’13. New York, NY, USA: ACM, 2013, pp. 31–36.

[KZK+12] S. Khemmarat, R. Zhou, D. Krishnappa, L. Gao, and M. Zink, Watching user generated videos with prefetching, Signal Processing: Image Communication, Vol. 27, No. 4, pp. 343 – 359, 2012, modern Media Transport – Dynamic Adaptive Streaming over {HTTP} (DASH). http://www.sciencedirect.com/science/article/pii/S0923596511001342

[L13] W. Luttrell, Only The Buy-Side Can Solve Our Fraud Problem. http://www.adexchanger.com/data-driven-thinking/only-the-buy-side-can-solve-our-fraud-problem/, 2013.

[LAH07] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. ACM TWEB, 2007.

[LB10] D. H. Lee and P. Brusilovsky, Social networks and interest similarity: The case of citeulike, in Proc. of HT, 2010, pp. 151–156.

[LBH80] W. J. Lonner, J. W. Berry, and G. H. Hofstede. Culture’s consequences: International differences in work-related values. University of Illinois at Urbana-Champaign’s Academy for Entrepreneurial Leadership Historical Research Reference in Entrepreneurship, 1980.

[LBK09] J. Leskovec, L. Backstrom, and J. Kleinberg. Meme-tracking and the dynamics of the news cycle. In SIGKDD, 2009.

[LC12] C. Lee and P. Cunningham. The geographic flow of music. In ASONAM, 2012.

[LCW10] K. Lee, J. Caverlee, and S. Webb, Uncovering social spammers: Social honeypots + machine learning, in In SIGIR, 2010.

[LG10] K. Lerman and R. Ghosh. Information contagion: An empirical study of the spread of news on digg and twitter social networks. ICWSM, 2010.

[LGK12] K. Lewis, M. Gonzalez, and J. Kaufman, Social selection and peer influence in an online social network, Proceedings of the National Academy of Sciences, vol. 109, no. 1, pp. 68–72, 2012.

[LGL08] Y. Liu, Y. Guo, and C. Liang, A Survey on Peer-to-Peer Video Streaming Systems, Peer-to-Peer Networking and Applications, vol. 1, no. 1, pp. 18–28, 2008.

[LGR+12] J. Lehmann, B. Goncalves, J. J. Ramasco, and C. Cattuto. Dynamical classes of collective attention in twitter. In Proceedings of the 21st international conference on World Wide Web, pages 251–260. ACM, 2012.

[LH08] J. Leskovec and E. Horvitz, Planetary-scale views on a large instant-messaging network, in Proc. of WWW, 2008, pp. 915–924.

[LNK05] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins, Geographic routing in social networks, PNAS, vol. 102, no. 33, pp. 11 623–11 628, 2005.

[LPN+11] K. Lee, D. Palsetia, R. Narayanan, M. Patwary, A. Agrawal, and A. Choudhary. Twitter trending topic classification. In ICDMW, 2011.

[LSW+12] Z. Li, H. Shen, H. Wang, G. Liu, and J. Li, Socialtube: P2P-assisted video sharing in online social networks, in INFOCOM, 2012 Proceedings IEEE, March 2012, pp. 2886–2890.

[LWB+09] W. Liang, R. Wu, J. Bi, and Z. Li, PPStream characterization: Measurement of P2P live streaming during Olympics, in IEEE Symposium on Computers and Communications, 2009.

[LWC12] R. Li, S. Wang, and K. C.-C. Chang, Multiple location profiling for users and relationships from social network and content. PVLDB, vol. 5, no. 11, pp. 1603–1614, 2012.

[LWD12] R. Li, S. Wang, H. Deng, R. Wang, and K. C.-C. Chang, Towards social user profiling: Unified and discriminative influence model for inferring home locations, in KDD, 2012, pp. 1023–1031.

[LZX+12] Z. Li, K. Zhang, Y. Xie, F. Yu, and X. Wang, Knowing your enemy: Understanding and detecting malicious web advertising, in Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS ’12, (New York, NY, USA), pp. 674–686, ACM, 2012.

[M04] M. Mitzenmacher. A brief history of generative models for power law and lognormal distributions. Internet mathematics, 1(2):226–251, 2004.

[MAA07] A. Metwally, D. Agrawal, and A. El Abbadi, Detectives: Detecting coalition hit inflation attacks in advertising networks streams, in Proceedings of the 16th International Conference on World Wide Web, WWW ’07, (New York, NY, USA), pp. 241–250, ACM, 2007.

[MAC13] A. Mansy, M. Ammar, J. Chandrashekar, and A. Sheth, Characterizing client behavior of commercial mobile video streaming services, in Proceedings of Workshop on Mobile Video Delivery, ser. MoViD’14. NY, USA: ACM, 2013, pp. 8:1–8:6.

[MCS+12] G. Magno, G. Comarela, D. Saez-Trumper, M. Cha, and V. Almeida. New kid on the block: Exploring the google+ social graph. In ACM IMC, 2012.

[Med13] Die Medienanstalten, Digitisation 2013 -Broadcasting and the Internet -Thesis, Antithesis, Synthesis? Tech. Rep., 2013.

[MGF13] R. Motamedi, R. Gonzalez, R. Farahbakhsh, A. Cuevas, R. Cuevas, and R. Rejaie, Characterizing group-level user behavior in major online social networks. Technical report available at: http://mirage.cs.uoregon.edu/pub/CIS-TR-2013-09.pdf, 2013.

[MMG07] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhat- tacharjee, Measurement and analysis of online social networks, in Proc. of IMC, 2007, pp. 29–42.

[MPG+11] B. Miller, P. Pearce, C. Grier, C. Kreibich, and V. Paxson, What's clicking what? techniques and innovations of today's clickbots, in Proceedings of the 8th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, DIMVA'11, (Berlin, Heidelberg), pp. 164–183, Springer-Verlag, 2011.

[MSC01] M. McPherson, L. Smith-Lovin, and J. M. Cook, Birds of a feather: Homophily in social networks, Annual Review of Sociology, vol. 27, no. 1, pp. 415–444, 2001.

[MVG10] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, You are who you know: inferring user profiles in online social networks, in Proc. of WSDM, 2010, pp. 251–260.

[NSS10] E. Nygren, R. Sitaraman, and J. Sun, The Akamai Network: A Platform for High-performance Internet Applications, ACM SIGOPS Operating Systems Review, vol. 44, no. 3, pp. 2–19, 2010.

[NTL13] NLTK modules for similarity. http://www.nltk.org/api/nltk.metrics.html. 2013

[O65] E. Ostgaard. Factors influencing the flow of news. Journal of Peace Research, 2(1):39–63, 1965.

[OB05] J. Gama Oliveira and Albert-Laszlo Barabasi. Human dynamics: Darwin and Einstein correspondence patterns. Nature, 437(7063):1251–1251, 2005.

[OFS14] Online Fraud Stats. http://www.ocalasmostwanted.com/online_fraud_stats.htm.

[PLR04] T. Peng, C. Leckie, and K. Ramamohanarao, Proactively detecting distributed denial of service attacks using source ip address monitoring., in NETWORKING (N. Mitrou, K. P. Kontovasilis, G. N. Rouskas, I. Iliadis, and L. F. Merakos, eds.), vol. 3042 of Lecture Notes in Computer Science, pp. 771–782, Springer, 2004.

[PMV12] T. Pontes, G. Magno, M. Vasconcelos, A. Gupta, J. Almeida, P. Kumaraguru, and V. Almeida, Beware of what you share: Inferring home location in social networks, in ICDM Workshop, 2012, pp. 571–578.

[PSF+07] M. P. Collins, T. J. Shimeall, S. Faber, J. Janies, R. Weaver, M. De Shon, and J. Kadane, Using uncleanliness to predict future botnet addresses, in Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, IMC '07, (New York, NY, USA), pp. 93–104, ACM, 2007.

[PVA12] T. Pontes, M. Vasconcelos, J. Almeida, P. Kumaraguru, and V. Almeida, We know where you live: privacy characterization of foursquare behavior, in UbiComp, 2012, pp. 898–905.

[RCC+13] Roberto Gonzalez, Ruben Cuevas, Angel Cuevas, and Carmen Guerrero. Understanding the locality effect in Twitter: measurement and analysis. Personal and Ubiquitous Computing, 2013.

[RF06] A. Ramachandran and N. Feamster, Understanding the network-level behavior of spammers, SIGCOMM Comput. Commun. Rev., vol. 36, pp. 291–302, Aug. 2006.

[RJR12] R. Dey, Z. Jelveh, and K. Ross, Facebook users have become much more private: A large-scale study, in PERCOM Workshop, 2012, pp. 346–352.

[RKH14] J. Rückert, T. Knierim, and D. Hausheer: Clubbing with the Peers: A Measurement Study of BitTorrent Live. In: 14th IEEE International Conference on Peer-to-Peer Computing, September 2014. (Best Paper Award IEEE P2P 2014)

[RM14] K. Ryoo and S. Moon, Inferring twitter user locations with 10 km accuracy, in WWW Companion, 2014, pp. 643–648.

[RMK11] D. M. Romero, B. Meeder, and J. Kleinberg. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter. In WWW, 2011.

[S08] C. Shirky. Here comes everybody: The power of organizing without organizations. Penguin, 2008.

[S10] B. Schwartz, Report: Click Fraud Rate for Q2 2010 28.9%. http://searchengineland.com/report-click-fraud- rate-for-q2-2010-28-9-45838, 2010.

[SAM12] F. Soldo, K. Argyraki, and A. Markopoulou, Optimal source-based filtering of malicious traffic, IEEE/ACM Trans. Netw., vol. 20, pp. 381–395, Apr. 2012.

[San13] Sandvine, Fall 2013 Global Internet Phenomena Report, 2013.

[SCL13] D. Saez-Trumper, C. Castillo, and M. Lalmas. Social media news communities: Gatekeeping, coverage, and statement bias. In Proceedings of the 22Nd ACM International Conference on Conference on Information; Knowledge Management, CIKM '13, pages 1679–1684, 2013.

[SEL] Selenium webdriver. http://docs.seleniumhq.org/projects/webdriver/.

[SFM+14] P. Singer, F. Flock, C. Meinhart, E. Zeitfogel, and M. Strohmaier. Evolution of reddit: From the front page of the internet to a self-referential community? In Proceedings of the Companion Publication of the 23rd International Conference on World Wide Web Companion, WWW Companion '14, pages 517–522, 2014.

[SHM02] S. Staniford, J. A. Hoagland, and J. M. McAlerney, Practical automated detection of stealthy portscans, J. Comput. Secur., vol. 10, pp. 105–136, July 2002.

[SIN01] A. Singhal, Modern Information Retrieval: A Brief Overview. IEEE Data(base) Engineering Bulletin 24 (2001), 35–43.

[Sit13] R. K. Sitaraman, Network Performance: Does It Really Matter To Users and by How Much? in IEEE COMSNETS, 2013.

[SKV10] G. Stringhini, C. Kruegel, and G. Vigna, Detecting spammers on social networks, in Proceedings of the 26th Annual Computer Security Applications Conference, ACSAC '10, (New York, NY, USA), pp. 1–9, ACM, 2010.

[SMM+11] S. Scellato, C. Mascolo, M. Musolesi, and J. Crowcroft. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In WWW, 2011.

[SMS14] Solve Media Survey. http://news.solvemedia.com/post/ 74832974631/solve-media-bot-survey-2014, 2014.

[Sol14] A. Solheim, DragonWave, Microwave back-haul radios meet the evolving traffic challenge, February 2013. http://mobiledevdesign.com/learning-resources/microwave-backhaul-radios-meet-evolving-traffic-challenge

[SOM10] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users: real-time event detection by social sensors. In WWW, 2010.

[SQ] Squid proxy server. www.squid-cache.org/.

[SRM09] E. Sun, I. Rosenn, C. Marlow, and T. M Lento. Gesundheit! modeling contagion through facebook news feed. In ICWSM, 2009.

[SSS+12] D. Schioberg, S. Schmid, F. Schneider, S. Uhlig, H. Schioberg, and A. Feldmann. Tracing the birth of an osn: Social graph and profile analysis in google+. In ACM WebSci, 2012.

[SSZ+11] B. Stone-Gross, R. Stevens, A. Zarras, R. Kemmerer, C. Kruegel, and G. Vigna, Understanding fraudulent activities in online ad exchanges, in Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC ’11, (New York, NY, USA), pp. 279–294, ACM, 2011.

[STA14] Leading internet multimedia portals in the United States in August 2014, based on market share of visits. http://www.statista.com/statistics/266201/us-market-share-of-leading-internet-video-portals/.

[STCL13] D. Saez-Trumper, C. Castillo, and M. Lalmas. Social media news communities: gatekeeping, coverage, and statement bias. In CIKM. ACM, 2013.

[STE14] M. A. STELZNER, How marketers are using social media to grow their businesses, 2014.

[SYC08] N. Sastry, E. Yoneki, and J. Crowcroft. Buzztraq: predicting geographical access patterns of social cascades using social networks. In ACM EuroSys, 2009.

[T14] Z. Tufekci. The medium and the movement: Digital tools, social movement politics, and the end of the free rider problem. Policy & Internet, 6(2):202–208, 2014.

[TTAPI] Trending Topics API Request. https://dev.twitter.com/rest/reference/get/trends/place.

[TWAPI] Twitter API Documentation. https://dev.twitter.com/.

[TWB] The World Bank. http://www.worldbank.org/.

[UKB11] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The anatomy of the facebook social graph. arXiv preprint arXiv:1111.4503, 2011.

[V14] S. Vranica, A ‘Crisis’ in Online Ads: One-Third of Traffic Is Bogus. http://online.wsj.com/news/articles/SB10001424052702304026304579453253860786362, 2014.

[VBS] S. Venkataraman, A. Blum, D. Song, S. Sen, andO. Spatscheck, Tracking dynamic sources of malicious activity at internet-scale.

[VGL+07] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, Measurement and Modeling of a large-scale Overlay for Multimedia Streaming, in ACM QSHINE, 2007.

[VOD+06] A. Vazquez, J. Gama Oliveira, Z. Dezso, K. Goh, I. Kondor, and A. Barabasi. Modeling bursts and heavy tails in human dynamics. Physical Review E, 73(3):036127, 2006.

[VSH+13] A. B. Vieira, A. P. C. da Silva, F. Henrique, G. Goncalves, and P. de Carvalho Gomes, SopCast P2P live streaming: live session traces and analysis, in ACM Multimedia Systems Conference on (MMSys), 2013.

[VWB+13] F. Viegas, Ma. Wattenberg, J. Hebert, G. Borggaard, A. Cichowlas, J. Feinberg, J. Orwant, and C. Wren. Google+ ripples: a native visualization of information flow. In WWW, 2013.

[W00] H. D. Wu. Systemic determinants of international news coverage: A comparison of 38 countries. Journal of Communication, 2000.

[W02] I. Wallace. The global economic system. Routledge, 2002.

[W03] H. D. Wu. Homogeneity around the world? Comparing the systemic determinants of international news flow between developed and developing countries. International Communication Gazette, 2003.

[W45] F. Wilcoxon. Individual comparisons by ranking methods. Biometrics bulletin, pages 80–83, 1945.

[W93] I. Wallerstein. The world-system after the cold war. Journal of Peace Research, 1993.

[W98] H. D. Wu. Investigating the determinants of international news flow a meta-analysis. International Communication Gazette, 60(6):493–512, 1998.

[WBL09] D. Richard, E. Watersa, A. L. Burnettb, and J. Lucasb, Engaging stakeholders through social networking: How nonprofit organizations are using facebook. Public Relations Review, Elsevier (2009).

[WIL45] F. Wilcoxon, Individual comparisons by ranking methods. Biometrics bulletin (1945), 80–83.

[WRR+14] M. Wichtlhuber, B. Richerzhagen, J. Rückert, and D. Hausheer, TRANSIT: Supporting Transitions in Peer-to-Peer Live Video Streaming, in IFIP NETWORKING, 2014.

[WSL+12] T. Wu, K. D. Schepper, W. V. Leekwijck, and D. D. Vleeschauwer, Reuse time based caching policy for video streaming, in CCNC, 2012, pp. 89–93

[WSS14] C. Wagner, P. Singer, and M. Strohmaier. Spatial and temporal patterns of online food preferences. In Proceedings of the companion publication of the 23rd international conference on World wide web companion, pages 553–554. International World Wide Web Conferences Steering Committee, 2014.

[WT12] D. Wilkinson and M. Thelwall. Trending twitter topics in english: An international comparison. Journal of the American Society for Information Science and Technology, 63(8):1631–1646, 2012.

[WWE13] WWE top 100 million facebook fans at wrestlemania. http://corporate.wwe.com/news/2013/2013_04_10.jsp.

[WYS12] M. Wysocki, The Role of Social Media in Sports Communication: An Analysis of NBA Teams Strategy. Tech. rep., 2012. http://www.american.edu/soc/communication/upload/Capstone-Wysocki.pdf.

[WZX+10] Y. Wu, C. Zhou, J. Xiao, J. Kurths, and H. J. Schellnhuber. Evidence for a bimodal distribution in human communication. Proceedings of the national academy of sciences, 107(44):18803–18808, 2010.

[XNR10] R. Xiang, J. Neville, and M. Rogati, Modeling relationship strength in online social networks, in Proc. of WWW, 2010, pp. 981–990.

[XYA+08] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, Spamming botnets: Signatures and characteristics, SIGCOMM Comput. Commun. Rev., vol. 38, pp. 171–182, Aug. 2008.

[YF09] B. Yu and H. Fei. Modeling social cascade in the flickr social network. In IEEE FSKD, 2009.

[YHG11] C. Yang, R. C. Harkreader, and G. Gu, Die free or live hard? empirical evaluation and new design for fighting evolving twitter spammers, in Proceedings of the 14th International Conference on Recent Advances in Intrusion Detection, RAID'11, (Berlin, Heidelberg), pp. 318–337, Springer-Verlag, 2011.

[YW10] S. Ye and S. F. Wu. Measuring message propagation and social influence on twitter.com. In Social informatics, pages 216–231. Springer, 2010.

[Z14] E. Zuckerman. New media, new civics? Policy & Internet, 6(2):151–168, 2014.

[ZCL+13] M. Zhao, A. Chen, Y. Lin, and A. Haeberlen, Peer-Assisted Content Distribution in Akamai NetSession, in ACM IMC, 2013.

[ZG07] C.-N. Ziegler and J. Golbeck, Investigating interactions of trust and interest similarity, Decision Support System, vol. 43, no. 2, pp. 460– 475, Mar. 2007.

[ZH12] X. Zhang, H. Hassanein, A Survey of Peer-to-Peer Live Video Streaming Schemes -An Algorithmic Perspective, Computer Networks, vol. 56, no. 15, pp. 3548–3579, 2012.

[ZP11] C. M. Zhang and V. Paxson, Detecting and analyzing automated activity on twitter, in Proceedings of the 12th International Conference on Passive and Active Measurement, PAM'11, (Berlin, Heidelberg), pp. 102–111, Springer-Verlag, 2011.

[ZSF+11] A. Zubiaga, D. Spina, V. Fresno and R. Martınez. Classifying trending topics: a typology of conversation triggers on twitter. In CIKM, 2011.

[ZZY13] J. Zhang, R. Zhang, Y. Zhang, and G. Yan, On the impact of social botnets for spam distribution and digital-influence manipulation, in Communications and Network Security (CNS), 2013 IEEE Conference on, pp. 46–54, Oct 2013.

ACRONYMS

ACC	Accuracy
AED	Average Error Distance
CDF	Cumulative Distribution Function
CP	Correct Predictions
CSV	Comma-Separated Values
CUTV	Catch-Up Television
D2D	Device-to-Device
DLNA	Digital Living Network Alliance
EC	Exposure coefficient
ED	Error Distance
EP	Exposure probability
F	Friend
FB	Facebook
FLI	Friend Location Indication
GB	Gigabyte
GGSN	GPRS Support Node
GPS	Global Positioning System
G+	Google+
HSDPA	High Speed Downlink Packet Access
HT	Hometown
ID	Identifier
KDE	Kernel Density Estimation
LA	Location Available
LA-FLI	Location available Friend Location Indication
LCC	Largest Connected Component
LN-FLI	Location non-available Friend Location Indication
LN	Location Non-Available
FSN	Federated Social Networks
LTE	Long Term Evolution
NAT	Network Address Translator
OSN	Online Social Network
PFLI	Profile and Friend Location Indication
PLI	Profile Location Indication

QoE	Quality of Experience
R	Average Reach
RMSE	Root Mean Squared Error
TCP	Transmission Control Protocol
TR	Total Reach
TT	Trending Topic
TW	Twitter
UGC	User Generated Content
UP	Useless Predictions
WE	Work and Education