A Socially Aware ISP-friendly Mechanism for Efficient Content Delivery

George Darzanos, Ioanna Papafili, George D. Stamoulis
Department of Informatics, Athens University of Economics and Business, Athens, Greece
{ntarzanos, iopapafi, gstamoul}@aueb.gr

Abstract—**The ongoing growth of the Internet traffic is largely due to video delivery platforms as well as to online social networks (OSN), where both popular and long-tail content, such as user-generated content (UGC), is shared. UGC is not dealt with effectively by the traditional approaches of Web caches and CDNs, as it has different demand patterns: it is more likely to be exchanged within a local geographic region and has a more even popularity distribution with fewer popular objects. In this work, we propose an ISP-friendly mechanism for enhancing content delivery exploiting social information extracted from OSNs (e.g. social relationships, common interests and locality of content exchange), as a new 'source' of meta-information to characterize and predict the end-user's behavior. The basic components of our mechanism called SEConD are: (i) a socially-aware proxy server inserted in a local geographic region (e.g. an AS) to orchestrate content distribution, (ii) socially-aware messaging overlays employed to trigger video prefetching, (iii) content-based P2P overlays employed to perform video streaming in each region and (iv) a two-level caching strategy both in the socially-aware proxy server and in the OSN user's device (UD) whenever online. We also develop an evaluation framework to simulate the generation of content in the environment of an OSN, in order to evaluate our mechanism and compare it with other approaches in the literature. The evaluation results show that our mechanism improves users' Quality of Experience (QoE) and simultaneously, reduces traffic in potentially expensive inter-domain links, as well as the origin content server contribution. Thus, the SEConD mechanism can lead to benefits for all involved stakeholders, i.e., ISPs, OSNs, CDNs, and end-users. The proposed mechanism is widely applicable, since it is deployable by ISPs, OSNs, as well as CDNs distributing content on behalf of an OSN provider.**

Index Terms — **socially-aware traffic management, video streaming, content placement, caching, QoE.**

I. Introduction

Online Social Network (OSN) are highly popular, as more and more people are using OSNs to keep in touch with their acquaintances, to entertain themselves etc. Thus, the data exchanged over OSNs represent a significant fraction of Internet traffic globally. Today, OSNs have changed the way we use the Internet, since everyday a huge amount of content are being created, stored initially, and shared at the edges of the network. Only, a very small percentage of the content shared in OSNs has high popularity. Most of the content is created by users, hence is called User Generated Content (UGC) and has a long-tailed nature with fewer popular objects and a large number of low popularity objects.

Video sharing in OSNs, is the main contributor of the traffic created by them. Videos in OSN are distributed through friends, by users' watch and share actions. Currently, Facebook has become the second largest video viewing platform after Google websites. Most OSN providers either deploy their own CDN following a client/server architecture to distribute videos, or assign the delivery of videos to third-party CDNs. Such solutions are costly primarily in terms of bandwidth but also in terms of storage in order to achieve high QoE for the users [1]. Moreover, traditional web-caching schemes, CDNs, and P2P assisted video sharing systems cannot deal efficiently with such content, because they don't take social relationships into account. Consequently, we identify the need for a scalable content distribution system for OSNs to efficiently deliver the content, achieving high QoE for users, while minimizing the operating costs of ISPs and OSNs providers. Recently, a variety of mechanisms have been proposed, which are addressing the efficient delivery of UGC by exploiting information such as: social relationships, interest similarities with respect to content, read patterns of OSN users, time-zone difference and locality of demand for OSN-published content. However, as will be seen in Section II, most of them are ineffective in terms of either inter-domain traffic costs or QoE for the end user, or scalability.

In this paper, we propose an ISP-friendly Socially-aware mechanism for Enhancing Content Delivery (SEConD); specifically, we deal with the delivery of videos as a case study. The proposed mechanism is socially-aware as it exploits social relationships, interest similarities with respect to content and locality of exchange of OSN content. SEConD addresses efficient delivery of both popular and long-tailed content in order to reduce the associated costs of ISPs (hence, it is ISP-friendly) and OSN while maintaining high QoE of OSN users.

SEConD is innovative in several aspects. First, based on social information and videos' classification into interest categories, multiple socially-aware messaging overlays are created per user, in each of which he can send demand indications to his friends for specific content items, in order to proactively store the prefix (first chunk) of those items. Additionally, a socially-aware proxy server (SPS) is employed in each Autonomous System (AS) in order to: orchestrate the formation of messaging overlays, to operate as P2P tracker for local content-based P2P overlays, and to achieve high traffic localization, by caching content to assist in sharing, when the local P2P is not adequate. Finally, SEConD employs a novel caching strategy, based on the demand patterns of OSNs rather than on general popularity of content. In Section II, we compare SEConD in detail to other socially aware mechanisms, most notably with those of [2] and [3] with which our mechanisms has some common features and several differences both in its objectives and in its approach.

The rest of this paper is organized as follows: in Section II, we provide an overview of the literature on content distribution mechanisms exploiting social information to enhance content delivery. In Section III, we describe the SEConD mechanism and its constituent elements. In Section IV, we specify a framework to simulate a social environment to evaluate our mechanism, we present some implementation details and we describe the evaluation setup. In Section V, we present our evaluation results and assess our proposed mechanism.

II. Related work

There are several works in the literature dealing with the delivery of content published in OSNs, either popular or long-tailed. In this section, we provide an overview of related mechanisms. First, we refer to two studies addressing the nature of the content published and distributed through OSNs. In [4], the authors argue that OSN websites can reveal valuable information that can be used by CDNs to improve caching and pre-fetching performance, and explain the relevant challenges. Moreover, [5] studies the correlation of patterns of propagation of video links in the microblogging eco-system with video popularity in the video sharing site. Also, [5] designs a neural network-based learning approach to predict the potential viewers of different videos and their geographic distribution, to deploy a proactive video sharing system.

Next, we overview some interesting approaches for content delivery over OSNs. A centralized approach called Tailgate is introduced in [6]. Tailgate uses social information and meta-information derived from OSNs, such as social relationships, regularities in read access patterns, and time-zone differences for predicting where and when the content will likely be consumed, in order to push the content wherever necessary before it is needed. Tailgate particularly addresses long-tail content efficiently and selectively distributes it across globally spread PoPs, while lowering bandwidth costs and improving QoE. Tailgate optimizes the traffic, in the level of inter-datacenter communication, by proactively pushing the content to PoPs close to potential viewers. On the other hand, SEConD employs messaging overlays to alert potential viewers of a content, in order to pull only its prefix. Due to this prefetching, SEConD achieves higher QoE for its users and scales better than Tailgate because of the local P2P overlays. Nevertheless, Tailgate and SEConD can be combined together.

SocialTube [2] and WebCloud [3] employ user-assisted content distribution techniques. WebCloud is a content delivery system for OSNs, which operates by repurposing users' web browsers to serve content to other users. In particular, WebCloud aims to serve users' requests by their friends in the OSN, instead of using directly the OSN server. Additionally, the authors of [3] state that WebCloud aims to keep the content exchange between two users within the same ISP and geographic region, in order to reduce the costs of both the OSN and the ISP. To achieve this, WebCloud introduces middle-boxes called redirector proxies, each of which is responsible to determine if any other online local user has the requested content. If so, a connection is established between these two users who eventually exchange the content. Should no local user have the content, the browser fetches the content from the OSN. Note that WebCloud employs caching at the user-clients and aims to achieve traffic localization by promoting end-to-end communication between users; this may result in QoE degradation for delay-sensitive content, e.g. video. On the other hand, SEConD comprises prefetching as well as the use of an SPS to act as an extra cache, both to achieve proactively traffic localization and to avoid any performance degradation. At the same time, SEConD also retains certain nice properties inherent to P2P overlays such as scalability and robustness.

In [2], the impact of social distance on video viewing patterns is investigated together with the correlation between user interests and video viewing patterns which is explored, by classifying videos in interest groups based on crawled Facebook and YouTube data. Based on the derived social observations, SocialTube is proposed. This is a peer-assisted video sharing system that explores social relationships and similarity of video interests among users in OSNs in order to create a P2P overlay with friendship-and- interest-based clusters. The SocialTube mechanism uses this overlay for efficient prefetching and video streaming, also in conjunction with buffer management leads to improvement of both the QoE of users and the system's scalability over current P2P video sharing techniques. To compare it with our mechanism, we note that SocialTube does not consider the IP network topology, and thus disregards the impact of the content distribution to the transit costs of the ISPs. Moreover, Socialtube constructs static P2P swarms for video delivery, considering only social relationships and interest similarities, but not the actual demand for specific content items. Thus, it generates unnecessarily many and large swarms, but each peer contributes to only a few of them. Moreover, each peer simultaneously participates in multiple swarms, thus resulting in increased management overhead. On the other hand, SEConD employs IP network topology information (*network awareness*) and generates small and local, yet content-centric P2P swarms, achieving higher efficiency and lower inter-domain traffic (*content awareness*). We follow a similar approach for the selection of potential viewers, but SocialTube does not use messaging overlays and performs a push based prefetching that gives rise to redundancy.

Finally, an interesting approach is the peer-assisted CDN of Akamai (NetSession [7]). In some cases, Akamai's edge servers are co-located (or peered) with ISPs and thus a high degree of traffic locality can be achieved. Consequently, [7] is an approach for content delivery that can handle effectively the popular content. SEConD, can also achieve a high degree of locality, but is innovative compared to [7] as it follows a caching strategy based on OSN's demand and thus can handle *both popular and long-tail content.* The latter is a large fraction of Internet traffic nowadays [6]. Also, contrary to [7], SEConD exploits social relationships to perform prefetching to enhance users QoE. Even in case where NetSession is already in place, SEConD can still be useful for targeted prefetching and for handling efficiently UGC by means of a caching strategy based on social demand.

III. The Proposed Mechanism

We describe our approach referring to Facebook, rather to OSNs in general, since most of OSN follow a similar approach on video sharing. Users in Facebook tend to watch videos driven both by social relationships and primarily by interest in the content of the video [2]. Additionally, most of the viewers of a video uploaded/shared by a user (uploader) have been observed to be within two hops in social graph from the uploader and the main viewers of the video are followers of the uploader, as defined precisely below. Finally, the videos published locally to a region have been observed to be mainly consumed by users in this region [3]. At first, taking these into account, we perform a categorization of the friends of each uploader (within two social hops), based on the influence that the uploader has in each one of them, namely the percentage of his videos they watch. This categorization will allow us to handle each category of friends separately, and perform a more efficient and targeted prefetching. Then, we describe the use-cases of video viewing in Facebook. Finally, we describe our mechanism SEConD and specify its constituent elements.

A. Categorization of Friends into Viewers Profiles

Next, we extend the categorization presented in [2] and we also define explicit threshold-based criteria for each category based on the results obtained from the measurement study they conducted. In particular, for each uploader, we consider as: a) *Followers*: his 1-hop or 2-hops friends that watch over 80% of the videos he uploads. b) *Non-followers*: his 1-hop or 2-hops friends that watch less than 80% but more than 30% of the videos he uploads. c) *Other viewers*: his 1-hop or 2-hops friends that watch less than 30% but more than 20% of the videos he uploads. Note that if a friend of an uploader watches his videos, then this friend watches at least 20% of them [2]. The rest of the friends, are not considered as viewers at all.

B. Use-Cases Addressed

We address two major and frequent use-cases of video sharing in popular OSNs (Facebook as use case), namely:

1) Video hosted in Facebook server: A user uploads a video on his Facebook profile; the video is uploaded to and hosted by the OSN video server. Then, each viewer of the uploader can view the video directly from the OSN video server.

2) *Video hosted in third-party owned server*: A user copies the link of a video from a third-party owned site (origin), such as YouTube, which directs to a third-party owned server (step 1, Fig. 1), and shares this link on his wall in Facebook (step 2). Then, each viewer of that video clicks on the shared link in the OSN site (step 3), is redirected (step 4) to the origin server that hosts the video, and eventually watches the video (step 5). This is the most frequent use case of video sharing in OSNs. According to [2] it accounts for 86% of videos, where most of them are videos hosted by YouTube (80%).

In both of the above described cases, the video server of the OSN or of the video sharing website may belong to a CDN, responsible for the distribution of their content.

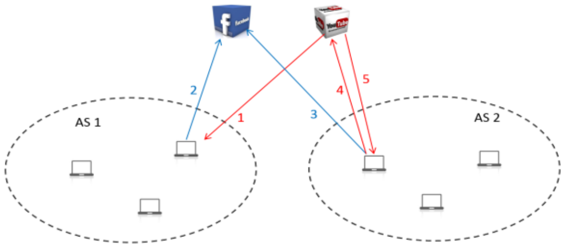

Fig. 1: Video hosted in YouTube's video server.

C. SEConD Mechanism

SEConD is a novel traffic management mechanism for enhancing content delivery over OSN exploiting social information derived by the OSN, as well as interest similarities and locality of exchange of OSN content. SEConD has been initially designed for enabling efficient video distribution. Nevertheless, it can be easily extended to provide the capability to efficiently handle any type of content shared in OSNs (e.g. photos). The main objectives of SEConD are the improvement of the QoE of OSN users in terms of decreased latency (eliminate stalling events), and the reduction of the inter-AS traffic, which we use as a proxy for transit inter-connection costs of the ISP. To this end, SEConD enables targeted prefetching of video prefixes based on social information, as well as caching and peer-assisted video delivery.

We assume that the OSN clients are properly adjusted in order to enable functionalities for the efficient operation of SEConD, like P2P capability of OSN clients, awareness of SPS, proper operation of different components of the system and communication between them, etc. Currently, these functionalities are not enabled, but OSN clients can be easily adjusted to support them transparently to users.

The basic *constituent elements* of SEConD are as follows:

1) ***Socially-aware messaging overlays*** are created and used to disseminate alert messages from the uploader of a video, to his friends that are potential viewers of this video. Each uploader is considered to have interest in specific video categories, and thus maintains a messaging overlay for each video category of his interest. Each messaging overlay, contains as potential viewers, all the followers of the uploader, and only the non-followers that are also interested in the respective video category. We choose to not include in messaging overlays the users of 'other viewers' category, since the percentage of the videos that they watch is rather low. Consequently, each uploader has a number of messaging overlays, equal to the number of video categories that he is interested in. Each messaging overlay is a bipartite graph, with edges only from the uploader to each potential viewer. Fig. 2 depicts an overlay constructed for an uploader and for one of his interest categories. When the uploader uploads a video uses the appropriate (based on interest category) messaging

overlay to alert potential viewers and trigger the pull-based prefetching of the video prefix, for QoE enhancement. Later, we describe how our prefetching algorithm taking advantage of messaging overlays. Finally, as observed in [2], 94% of the videos each user watches are at most from 4 video categories. Thus, for each uploader, we choose to create overlays only for his top 4 interest video categories. Also, based on total measurements of [2], the estimated average number of users participating in each overlay is no more than 130.

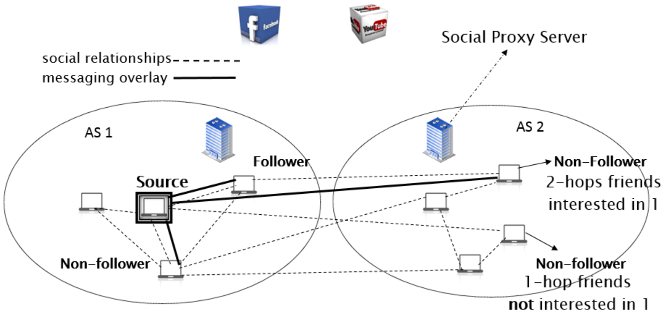

Fig. 2: The messaging overlay of source user for the interest category 1.

2) *The **Social Proxy Server** (SPS)* is located within each AS in order to localize the traffic generated by the activity of OSN users in this region. Each user of the OSN is considered to be aware of the SPS of his home AS, and thus can request videos and video prefixes from the SPS. Also, the SPS communicates with the servers of the OSN provider and the video streaming platform provider. The SPS can be controlled by either the OSN provider, the CDN, or the local ISP. The role of the SPS is diverse and important for the robustness of the system. The SPS: i) is responsible for the formation of messaging overlays and to keep them updated through monitoring, ii) responds to users' requests for video prefixes, received through the messaging overlays, by pushing the requested video prefixes, iii) caches the prefixes to serve future requests, iv) operates as local P2P Orchestrator (e.g. like a BitTorrent tracker) for local content-based swarms formed to perform the video streaming, v) in order to boost the users' QoE and localize traffic, the SPS caches each requested video by its local users. Therefore, the SPS is capable of participating in the P2P video delivery as a resourceful peer ([8]) when needed, and to serve as a proxy video server ([3]). The relevant criterion we introduce for the SPS participation is that the per user available upload bandwidth in the swarm is below the video bit rate. Of course, there are different strategies, metrics and thresholds that could be employed for the contribution of the SPS.

3) ***Local Content-based P2P overlays*** are created by the SPS for each video that is requested by the local OSN users. Whenever a user requests to watch a video, the SPS checks if there is already a local P2P swarm for this video. If there is, the SPS adds the requesting user in this swarm and stores the video in his cache (if not already). If there isn't, the SPS creates a swarm for this video and stores the video in his cache. So, in this case the new swarm includes only the user requesting the video and the SPS. The SPS in any case assists in sharing until the upload rate offered by other peers/users is adequate to achieve the desired QoE. Consequently, the users interested in watching a video are added by the SPS in the swarm of this video in order to simultaneously download and share the video (leechers). While, the users who have stored a video in their UD, are added in the corresponding swarm in order to assist the SPS in sharing and traffic localization (seeders). If two neighboring ASes have a peering agreement, then the P2P overlays for distribution of videos can be extended accordingly. Although the swarms created can be small, effective resource allocation is achieved.

*4) **Caching*** is important to SEConD for QoE enhancement, and for achieving high reduction both in the inter-AS traffic created by video delivery and in the contribution of the server hosting the video. In SEConD, we follow a two-level caching strategy: prefixes and videos are cached both in the SPS and in UD. Caching prefixes in UD aims to decrease latency by eliminating the video start-up delay (or stall time), while caching the videos themselves in UD aims to assist in P2P video sharing. On the other hand, caching prefixes and videos in the SPS is done mainly for traffic localization, as done in certain approaches for P2P traffic, such as in [8].

Due to the fact that the storage capacity of both the UD and the SPS is limited, we need to define caching policies in order to determine which prefix(es) or video(s) to replace, when upon a new arrival the cache is already full. For simplicity, we chose the *caching policy* described below:

a) Caching in SPS: When a new video prefix arrives, if the SPS cache is full, the oldest prefix is replaced. Also, for each video, the SPS maintains a counter. This is increased whenever the relevant prefix is pushed to a user and decreased when the prefix is viewed or deleted from some UD. The UDs are assumed to inform the SPS for such deletions. Thus, the higher the counter of a video is the higher the possibility for the SPS to get requests for this video is. Therefore, when a new video arrives, the two oldest videos in the SPS cache are considered and the one with the lowest counter is replaced.

b) Caching in UD: When a new video prefix (or video) arrives it replaces the oldest prefix (or video) in the UD.

Concerning the size of the SPS cache, for simplicity, we consider it proportional to the number of users that are connected to it. In particular, in our evaluation framework, we take that the total storage capacity equals the number of users that are connected to the SPS by the size of one video prefix plus one video, which is a rather strict assumption. In fact, the cache size is not required to scale linearly with the number of the users, rather a fixed yet relatively small value, e.g. a few tens or hundreds of GBs, can suffice to successfully assist peer-assist content delivery, as argued in [9].

Finally, we describe the steps of our **socially-aware pull-based prefetching algorithm**, as depicted in Fig. 3: 1) When an uploader uploads a video, he pushes an alert message to each user in the messaging overlay that corresponds to the interest category of this video. 2) After a user receives an alert message, he sends a request to his local SPS for the prefix of the video referred in the message. 3) When the local SPS receives a prefix request, if this is not already cached, then the SPS downloads the video from the video server hosting it.

4) The local SPS caches the prefix of the video and pushes it to the user who requested it. Finally, the user stores the prefix of the video in his UE. In Fig. 3, we present the steps of the prefetching algorithm for the case where the video is uploaded to a third-party owned server, e.g. a YouTube server.

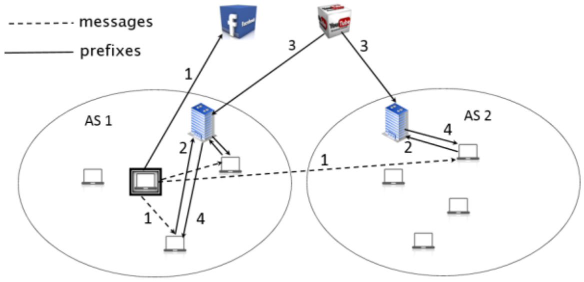

Fig 3: An example of the prefetching algorithm – The source node shares a video hosted in a third-party owned server. (Sequence numbers arrows.)

D. Discussion on SEConD Deployment and Benefits

SEConD is deployable by the OSN provider itself, or by ISPs, or by CDNs that operate complementarily to an OSN serving the content shared over it. In particular, an OSN (or CDN) provider can transparently deploy the mechanism by adjusting the OSN clients, in order to both achieve higher QoE for its users and reduce the workload of its video server.

Moreover, if an ISP deploys the mechanism, then social information should be acquired. In this case, the ISP has either to establish an agreement with the OSN provider in order to obtain that information, or to acquire it by crawling the OSN. In the latter case, the acquired information would be more limited. The deployment of SEConD by the ISP would again improve its users' QoE and at the same time result in savings on inter-AS traffic and on the associated charges. Of course, the ISP incurs the extra cost for deploying and running an SPS, which in case of owned infrastructure is expected to be much lower than the above savings. Note also that the OSN provider has the incentive to collaborate with an ISP in information exchange, since the SEConD leads to reduction of OSN server contribution and of the associated OSN operational costs.

Regarding the incentives of a CDN, which is commissioned by an OSN, to deploy the mechanism, it should be noted that a CDN takes care primarily of overall popular content. On the other hand, SEConD can also handle UGC that is popular within a group of users; this group is spotted by employing the social relations. Also, SEConD can achieve high scalability, taking advantage of P2P. See also discussion in the end of Section II on the relation of SEConD and NetSession [7].

IV. Evaluation Framework

In order to evaluate SEConD and its components and to compare it with SocialTube [2], we designed and implemented an evaluation framework to simulate a social environment. Our objective is to model the processes of posting and selection of videos by the OSN users. Thus, we employed observations in the literature regarding content delivery over OSNs, as well as users' behavior and interactions among them due to video viewing and sharing. Our assumptions are based on measurements studies conducted in real OSNs ([2], [10]-[18]).

A. Framework Components and Parameterization

In this section, we provide a detailed description of the components of our evaluation framework and their parameters. Furthermore, we assign values to certain parameters based on social environment observations.

1) Number of Interest Categories

Indeed, each video belongs to an interest category, while users have interest in some videos belonging in specific interest categories. We adopted a categorization similar to [10], where the authors present the various video categories and their distribution in YouTube. Note that we have considered in our framework two more categories additionally to the 17 YouTube's categories, due to the fact that the categorization of videos in [2] has been performed on the basis of 19 categories and we aim to compare our results to those of [2]. Therefore, we decided to add two more categories with a share of 0.5% in total video delivery for each one of them, while we reduce by 0.1% the top ten categories, i.e. those with the highest share.

2) Timing of Users' Activities

According to [11], it has been observed that around 66% of Facebook users are daily active users (DAUs). With the DAU parameter we refer to the average number of users are active in Facebook every day. Additionally, it has been noted that on the average, the users of Facebook spend daily 20 minutes in the website chatting, watching videos and viewing photos, posting comments or updating their status [11]. Moreover, in [12], every user is observed to spend on the average around 140 minutes every day in the Internet for different purposes. Finally, as reported in [13], the intensity of users' interaction in Facebook varies for different hours of the day. [13] also provides a detailed graph of users' interaction during a full day.

Based on the above information, we design a discrete-time event-driven simulation framework, where time is slotted in slots of 20 minutes. We assume that every day each user is active on the Internet for 7 20-minute slots, regardless his daily activity in Facebook. On the other hand, only 66% of users are active a given day in the OSN, and each such user is active only for one 20-minute slot within this day. To choose the exact set of users that are active in the OSN (and represent 66% of the total population), we make a random selection. The probability for each specific user to be selected is proportional to his weight; since, users with more friends are more likely to be active, user weights are defined according to the formula:

$$user\_weight = 1 + users'\_friends / 1000 .$$

Furthermore, in order to select the specific 20-minute slot a user is active in the OSN, we perform weighted random choice based on the information extracted by the activity graph in [13]. In the same way we choose the rest 6 slots where the user is active only in the Internet. We assume that during these 7 slots, where a user is active in the Internet, he is able to seed content that he has previously stored.

3) User characteristics and Viewer Categorization

As described in [2], all OSN users watch videos mostly from 4 out of the 19 video categories (94% of watched videos) and the rest 6% from all other categories. In our framework, we ignore the last possibility, for simplicity. Thus, to decide in which 4 categories a user is interested in, we used a weighted random choice and we chose 4 categories out of 19 total interest categories, using as weights the new percentages derived from the processing of the percentages presented in [11] (see subsection IV.A.1). The percentage of the videos a user watches from each one of his top 4 interest categories varies, since he may be interested in some categories more than others; this matter is treated later in the section.

Additionally, we assumed a specific number of ASes in the network layer, and we assigned to each AS a rank used to determine the population of the AS; i.e. ASes with higher rank, have more users. Then, in order to distribute the OSN users among the ASes, we used the Zipf distribution and assigned an AS id to each user. According to [14], during a month, Facebook users do manage to reach 35% of their friends with each post and 61% of their 1-hop friends. In our framework, we set the 61% of friends of each user as his monthly audience at 1-hop. As a result, we achieve to produce an average audience of 34% of 1-hop friends per post. This is in agreement with the social observations on datasets extracted by real OSNs in [14], while it verifies that our evaluation framework adequately approximates an actual OSN environment and reflects OSN users' behavior.

In SocialTube [2], it has been observed that 90% of the viewers of the videos of an uploader are within at most two social hops away from him, while the remaining 10% of viewers are in three hops or more. Especially, 70% of the viewers of the uploader are 1-hop friends of him, while 20% of viewers are 2-hops friends of him. On the average, 25% of the viewers of an uploader watch all videos that he has uploaded. Moreover, 33% of the viewers watch 80% or more of his videos and all of his viewers watch at least 20% of his videos. Also, they observed that 65% of non-followers and other viewers of an uploader are 1-hop friends of him. Also, the viewing activities of followers are primarily driven by social relations, while those of non-followers are driven mainly by interest. Based on these social observations and the categorization of friends in Subsection III.A, we specify the distribution of viewers (i.e., 61% of friends) of an uploader in one and two hops from him over the three categories of viewers, choosing from users with at least one common interest with him: a) *Followers:* 36.3% of viewers are characterized as 1- hop followers, while 2.2% of viewers are characterized as 2-hops followers, b) *Non-followers:* 40.6% of viewers are characterized as 1-hop non-followers, while 13.2% of viewers are characterized as 2-hops non-followers, c) *Other viewers:* 2.2% of viewers are characterized as 1-hop other viewers, while 6.5% of viewers are characterized as 2-hops other viewers.

4) *Videos and related interactions*

We created a hypothetical pool of videos in order to simulate a video platform like YouTube. We assigned popularity to each video using the Power Law distribution and an interest category by weighted random choice using as weights the percentages of each one of 19 interest categories. Thus, we create a number of videos with a distribution over 19 categories similar to YouTube and long-tailed regarding the popularity. Each user is assumed to have access to the videos published from his 1-hops friends, while viewers at 2 hops arise by means of sharing videos; this is a realistic assumption taking into account the privacy settings employed in OSNs such as Facebook. However, we assume that users watch videos related to their interests. As expected, videos of their top interests, as well as videos with highest popularity are more likely to be watched. Thus, the actual percentage of the videos a user watches from each category depends on his interest.

Next, we specify the number of videos watched per user. Since, each user is active in Facebook for 20 minutes on the average per day and considering that the average length of a video is 4 minutes [15], we assume that a user may watch from 1 to 5 videos in this 20-minute interval. Thus, the number of videos watched is chosen according to the uniform distribution. According to [16], 1 million links to external websites are being shared every 20 minutes in Facebook. Practically, this means that the total amount of links that are shared in Facebook every day is 15 times smaller than the total number of Facebook users. Also, most of these links were found to be mainly videos or pages containing videos. Furthermore, on the average, the videos uploaded and hosted in a Facebook server account for about 14% of all videos uploaded in Facebook. The remaining videos are hosted in other video platform, with 80% of them being hosted by YouTube [2]. Considering the above Facebook statistics, it is rather realistic to assume that the number of videos uploaded daily in our system equals to 5% of the total number of user in our system.

For each day, we decide which users will upload/share videos, by employing the Bernoulli distribution. That is, the users are chosen with uniformly randomly from the set of this day's active users. Additionally, each user can upload none, one or more videos, but only within the 20-minute slot that he is active in the OSN. According to [17], Facebook constitutes currently the second largest source for videos, and it generates about 11.8% of all referred video traffic in the Internet. Based on this observation, we thus assume that the number of videos that a user re-shares 11.8% of the total number of videos that he uploads. We characterize as 're-shares', the videos that a user watched from a post of one of his friends and then posted it on his profile, regardless in which server the video is hosted.

5) *Evaluation Metrics*

Next, we define a set of metrics of interest, which we consider important and we monitor during our simulations:

a) Inter/Intra AS traffic: We estimate the traffic generated by video distribution (including prefetching) both in the intra-AS and inter-AS links. Inter-AS traffic may lead to transit traffic charges, while intra-AS traffic affects the congestion created and can be considered as a proxy of the degradation of users' QoE. We also consider separately the traffic through peering links in some of our evaluations.

b) Contribution of server hosting the video: We estimate the percentage of traffic handled by the origin server e.g YouTube, where the video is hosted. We aim to achieve low contribution of the origin server to reduce the operational costs for the CDN or OSN video server platform and to avoid a bottleneck that can affect adversely the QoE of users.

c) (Overall) Caching accuracy of Social Proxy Server: We estimate the percentage of video prefixes or videos that had already been stored in the cache of the SPS when a user requested it. It is expected (and will be verified) that a higher caching accuracy of the SPS translates to lower contribution of the origin server where the video is hosted, and thus, to lower inter-AS traffic and potentially to lower transit charges.

d) Accuracy of prefetching: We estimate the percentage of video prefixes stored in a user UD when he requests to watch the corresponding video. High overall prefetching accuracy is expected to lead to zero or insignificant start up delay for the users, and consequently, high users' QoE.

e) Useless prefetching: By this term, we refer to the amount of video prefixes pushed and never used by the users that received them. A high number of useless prefixes expected to lead to some QoE deterioration due to bandwidth consumption and to low prefetching accuracy, due to the fact that these prefixes consume an amount of the local storage.

f) Redundant prefetching: this occurs when the same prefix is being pushed to a user by multiple sources, i.e. two or more of his friends. A high number of redundant prefixes is again expected to lead to some QoE deterioration.

B. Implementation Details and Evaluation Setup

In this section, we provide the implementation details for our evaluation framework, we discuss decisions made during the implementation, and we describe the exact framework setup for our evaluation.

We implemented our evaluation framework in MATLAB. As already mentioned, we divided the time of our system in 20-minute slots. However, in order to better handle the users' activities and to derive measurements of the various metrics introduced above, we divide every 20-minute slot in smaller slots of 4-minutes, i.e. the average time duration of a video shared over an OSN. We assume that each user is available to serve the local P2P overlay as leecher during a 4-minute slot when he is watching a video in Facebook, while each user is available to serve the local P2P overlay as seeder during every 20-minute slot during which he is online in the Internet. Note that we don't capture the dynamic nature of the messaging overlays created for each user, since we do not update them during the simulations. The estimation of the intra-AS traffic generated by a user watching a video, is based on the percentage of seeders and leechers (peers) that are active during this 4-minute slot and additionally, are located within the same AS or a peering AS. The estimation of inter-AS traffic is based on the percentage of peers that are located in remote ASes (except peering ones) combined with the contribution of the origin server where the video is hosted.

To capture the QoE of a user watching a video under the mechanism deployment, we use as proxy the per user available upload bandwidth in the swarm, as well as that from the SPS (or the origin server in case of SocialTube). Our main objective is to keep this available bandwidth for each user higher than the bit rate of a video being watched in order to assure high or at least adequate QoE level. In order to estimate inter- and intra-AS traffic and QoE, we also have to assign specific values of upload (UL) and download (DL) bandwidth to every user in our system. To this end, we employ three different bandwidth-access profiles, based on statistics presented in [18].

Finally, we present our assumptions on upload/share and watch actions of users. When a user uploads a video, he is assumed to select it from the video pool, by considering the interest categories and popularity values of the videos. When a user re-shares a video, he is assumed to select it randomly from his "watched" list. On the other hand, when a user wants to select a video to watch, first he is considered to select an available video from one of the three different groups of his friends, namely those of whom he is follower, those of whom he is non-follower, and those of whom he is just other viewer, according to a weighted random choice (follow 80%, non-follow 55%, other 15%). Then, the user selects a video based on his interest categories and on the popularities of the videos available. If there are no eligible videos in the selected group of users, then he proceeds with the next selected group.

Next, we describe the framework setup for our evaluation: First, we generate 3963 users (nodes) and create a social graph using the Facebook SNAP dataset [19]. Then, we distributed the users in 4 ASes of varying sizes using the Zipf distribution. Thus, the AS with id 1 has rank 1 and the highest number of users, while the AS with id 4 has rank 4 and the least users. Moreover, we created a pool of 9000 videos and we assigned to each video an interest category and a popularity value. Also, each video is taken to have a uniformly distributed random size from 20 to 30 MB and 330 Kbps bit-rate.

Furthermore, we set the cache size at the UD of each client equal to 300 MB, which can be considered as a rather low value taking into account the TBs of storage now available at low cost in users' UD. The cache size on each one of the four SPSs (one SPS per AS) is equal to the number of users assigned to the respective AS times the storage capacity for one prefix and one video, which is taken as 33 MB. Finally, the simulation lasted for 30 cycles corresponding to 30 days.

V. Evaluation Results

In this section, we present and discuss the evaluation results for the proposed mechanism, and we perform a comparison of SEConD to the mechanism of SocialTube presented in [2].

A. Prefetching Accuracy

We observed that both SEConD and SocialTube achieve *high prefetching accuracy*, namely around 88%. This is due to the fact that both mechanisms follow a similar approach in the selection of potential viewers where a prefix will be pushed. We also observed that, as in SocialTube, the prefetching accuracy under SEConD is higher for users that watched more videos and for those that received more prefixes. This can be exploited for fine-tuning SEConD in future extensions.

B. Inter-AS Traffic Reduction

One of the main targets of SEConD is to achieve reduction of inter-AS traffic generated due to video delivery. The results obtained show that our mechanism indeed achieves significant reduction of inter-AS traffic.

If we assume that ***prefetching*** in the current OSN video sharing system architecture, i.e., following a client-server architecture, generates daily a total of 100% inter-AS traffic, then SocialTube is found to generate 66% of inter-AS traffic, while SEConD only 12.6%. This high reduction is achieved due to the fact that in SEConD the users are pushing alert messages through the messaging overlays instead of pushing video prefixes, whose size is larger than that size of alerts. Then, the receivers of the alert messages can request any desired prefix from the local SPS. On the other hand, in SocialTube the prefixes are pushed from the users directly to all of their viewers in the clusters. Thus, in SEConD the prefix of each video is downloaded only once per AS, and thus the *redundant inter-AS traffic* due to the same prefix is eliminated.

Moreover, SEConD achieves high reduction of the ***total inter-AS traffic*** generated by the complete process of video delivery, including alerts and prefetching. Under SEConD, the total inter-AS traffic generated accounts for only 13% of the total inter-AS traffic under the client-server paradigm; namely, a *87% reduction* of total inter-AS traffic is achieved by SEConD. For SocialTube, a reduction of inter-AS traffic of 18% compared to client-server is attained. Fig. 4 illustrates the total inter-AS traffic generated by each mechanism within a specific full day. Clearly, SEConD achieves a high total reduction during the day, while it is very effective during the 'busy' hours although users' activity is higher, since more users are active and local P2P contribute more.

C. Third-party owned Server Contribution

We investigate the total ***contribution of the origin server*** where the video is hosted. SEConD achieves high reduction of the percentage of traffic handled by the origin video server. In the client-server architecture, the origin server contributes 100% of the video traffic. The relevant server's contribution in SocialTube drops to 55.3%, of the video traffic in the client-server case, while under SEConD it drops to 12.1% thereof. Practically, when SEConD is in place, inter-AS traffic is generated mainly due to the contribution of the origin video server, which is maintained at a low level.

Metrics	**ASes**			
	AS1	***AS2***	***AS3***	***AS4***
Number of users	1925	928	634	476
Proxy contribution	48%	61%	75%	79%
Proxy cache size	63GB	30GB	21GB	15.7GB
Proxy cache hit accuracy (videos/prefixes)	90/94 %	78/86 %	68/80 %	60/74 %
Origin server contribution	5%	14%	24%	31%

Table 1: Proxy server and SPS contribution w.r.t. the AS size.

D. AS Size vs Origin Server, SPS and P2P Contribution

Finally, we investigate the impact of the AS size on the contribution of the SPS and the origin video server under SEConD. In Table 1, we observe that as the size of the AS increases, the **contribution of SPS** decreases. For instance, we see that in the largest AS (i.e., AS1), the contribution of the SPS is 48%, and due to the very high (~90%) hit accuracy of the cache, the contribution of the origin server is just 5%. On the contrary, in the smaller AS (i.e., AS4), the contribution of the SPS is 79% and that of the origin server is up to 31%. This is explained by the fact that the resources of the local P2P overlay are lower for AS4, due to lower number of users, and thus SPS supports video distribution more frequently. Also, we observe that the **contribution of the origin** server decreases with the hit accuracy and the contribution of the SPS cache. Indeed, when a user requests a video from the local SPS, the SPS requests the video from the origin server, only if he doesn't have it cached.

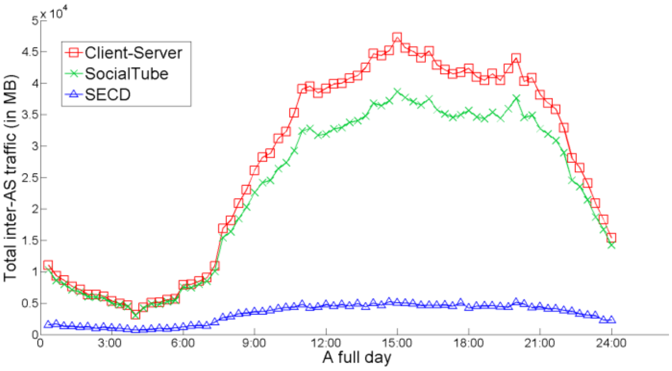

Fig. 4: Total inter-AS traffic during the day

Fig. 5 depicts the contribution of the SPS versus the **contribution of the local P2P overlay** for ASes of different sizes. The SPS contribution in smaller ASes is higher than that of P2P (see curves for AS4 in Fig. 5). We can also observe that the local P2P overlay will contribute more during the "busy" (or peak) hours, when more users are active and join the swarms, thus decreasing the workload of SPS and the origin server. Finally, by observing Table 1, we also notice that relatively less **SPS caching capacity** is needed in large ASes than in small ones, in order to achieve lower utilization of the origin server and also preserve the same (high) QoE level. Nevertheless, we observe that the SPS caching capacity that is needed in small ASes is low, e.g. 15.7 GB for a small AS of 476 users, which nowadays has low cost. For larger ASes, the inherent scalability of P2P results in reduced SPS contribution and thus indirectly in reduced contribution of the origin server. Thus, larger groups of users are self-sustained and need less SPS caching capacity.

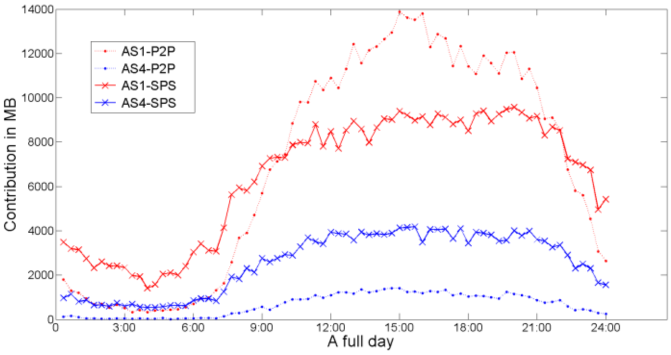

Fig. 5: SPS contribution vs local P2P contribution in different size ASes

E. Considering peering links

We also set up and ran additional experiments where we assumed a peering agreement between different pair of ASes each time. The evaluation results reveal that under a peering agreement both SocialTube and SEConD can be more beneficial in terms of total inter-AS traffic created, additionally SEConD can be more effective in terms of reducing the origin server contribution. The major factor that affects the level of benefit achieved is the size of ASes between whom the peering agreement is established. Thus, the bigger the sizes of the peering ASes the higher the improvement both mechanisms can achieve. In particular, when a peering link is in place, the *reduction* in the total inter-AS traffic of the client-server architecture achieved by SocialTube raises from 34% (see Subsection V.B) to 62-73% (depending on the size of the peering ASes), while the contribution of the origin server remains high at 55.3%. On the other hand, SEConD achieves higher utilization of the local P2P overlay when a peering link is in place, thus achieving even lower contributions of both the SPS (48-57%) and the origin server (7-8.5%), with the latter being equal to the total inter-AS traffic. Thus, when two ASes have a peering agreement, both mechanisms have improved performance, with SEConD still outperforming SocialTube.

VI. Conclusions and Future work

In this paper, we developed and evaluated a mechanism (SEConD) for ISP-friendly socially aware content delivery. SEConD was initially inspired by other mechanisms (mainly SocialTube [2] and WebCloud [3]), but employs several innovative ideas, such as the construction of socially aware messaging overlays, the use of socially-aware proxy servers (SPSs), and network topology awareness. According to our evaluation results, SEConD: a) enhances the QoE of OSN users both by achieving high overall prefetching accuracy and by maintaining an adequately high upload bandwidth within swarms, b) achieves high reduction of the inter-domain traffic, and thus, may result in reduced charges for transit inter-AS traffic, c) reduces the contribution of the video server of the OSN (CDN), as well as the relevant operational costs, i.e. mainly costs for bandwidth, and d) eliminates redundant prefix downloads, leading to reduction of traffic congestion within the AS. We also compared the performance of SEConD to SocialTube, since they are both suitable for video distribution. SEConD appears to outperform SocialTube in terms of the contribution of the origin server and of the total amount of inter-AS traffic generated due to the video distribution. The two mechanisms achieve similar prefetching accuracy, and thus similar levels of startup delay for watching a video.

There are several potential promising future extensions of SEConD. In particular, for videos being hosted in the OSN server, we can employ direct caching of the entire video in the SPS of the AS of the source node (from which it will be most likely requested) to achieve high QoE in the first downloads. We also plan to evaluate the monitoring component of SPS that is responsible to periodically update the messaging overlays according to policies based on the demand for videos, social relations, other user actions etc. Finally, alternative caching policies and different sizes of the cache of the SPS can be employed and evaluated, especially in the case of small ASes, where caching has proven to play a more significant role.

References

[1] Ager, Bernhard, et al. "Revisiting cacheability in times of user generated content." INFOCOM IEEE Conference on Computer Communications Workshops, 2010. IEEE.

[2] Li, Ze, et al. "Socialtube: P2P-assisted video sharing in online social networks." INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.

[3] Zhou, Fangfei, et al. "WebCloud: Recruiting social network users to assist in content distribution." Network Computing and Applications (NCA), 2012 11th IEEE International Symposium on. IEEE, 2012.

[4] Ruhela, Amit, et al. "Towards the use of online social networks for efficient Internet content distribution." Advanced Networks and Telecommunication Systems, 2011 IEEE 5th International Conference on. IEEE, 2011.

[5] Wang, Zhi, et al. "Guiding internet-scale video service deployment using microblog-based prediction." INFOCOM, 2012 Proceedings IEEE. IEEE, 2012.

[6] Traverso, Stefano, et al. "Tailgate: handling long-tail content with a little help from friends." Proceedings of the 21st international conference on WWW. ACM, 2012.

[7] Zhao, Mingchen, et al. "Peer-assisted content distribution in Akamai netsession." Proceedings of the 2013 conference on Internet measurement conference. ACM.

[8] Papafili, Ioanna, Sergios Soursos, and George D. Stamoulis. "Improvement of bittorrent performance and inter-domain traffic by inserting isp-owned peers." In Network Economics for Next Generation Networks, pp. 97-108. Springer Berlin Heidelberg, 2009.

[9] Papafili, Ioanna, and George D. Stamoulis. "A Markov model for the evaluation of cache insertion on peer-to-peer performance." *Next Generation Internet (NGI), 2010 6th EURO-NF Conference on*. IEEE, 2010.

[10] Yang, Weilong, and Zhensong Qian. "Understanding the Characteristics of Category-Specific YouTube Videos."

[11] Ellison, N. B., Steinfield, C., and Lampe, C.. The benefits of Facebook “friends:” Social capital and college students’ use of online social network sites. Journal of Computer‐Mediated Communication, 12(4), 1143-1168.

[12] http://www.nielsen.com/us/en/newswire/2009/top-u-s-web-brands-and-parent-companies-for-september-2009.html.

[13] Managing Your Facebook Community: Findings on Conversation Volume by Day of Week, Hour, and Minute, An Oracle White Paper, August 2012.

[14] Bernstein, M., et al. "Quantifying the invisible audience in social networks." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2013.

[15] http://www.sysomos.com/reports/youtube.

[16] http://www.statisticbrain.com/facebook-statistics/.

[17] http://www.longtailvideo.com/blog/22445/social-sharing-publish-your-videos-to-facebook-twitter-youtube.

[18] Cheng, Xu, and Jiangchuan Liu. "Nettube: Exploring social networks for peer-to-peer short video sharing." INFOCOM 2009, IEEE. IEEE, 2009.

[19] http://snap.stanford.edu/data/egonets-Facebook.html.