This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2011 proceedings

Utilizing Social Influence in Content Distribution Networks

Hossam Sharara
University of Maryland
College Park, MD
hossam@cs.umd.edu

Cedric Westphal, Svetlana Radosavac, and Ulas C. Kozat
DOCOMO USA Labs
Palo Alto, CA
{cwestphal,sradosavac,kozat}@docomolabs-usa.com

***Abstract*—Online social networks (OSNs) provide new means of disseminating information about applications and contents served by network providers. OSN members often reveal their usage information and opinion about applications and contents to their neighbors within their social networks. Consequently, sudden popularity and viral propagation of applications among OSN members can put significant burden on network resources and degrade network performance. Further, viral exchanges might propagate malicious applications and such propagation might need to be kept in check. Accordingly, we propose a novel content distribution architecture that controls the resource utilization within an operator's network by utilizing the existing social connections between users and building a model of information diffusion within the social network.**

Our method is based upon computing a reward function that takes into account the influence of users over each other in a given social network in terms of application adoption and content consumption. Based on this reward function and assuming that not only the users but also the network operator can limit the exposure of application usage and content consumption over the online social networks, we present algorithms to slow down or speed up the adoption/consumption of different applications/contents given the current state of both the physical network and the social network. We evaluate the effectiveness of this method over a Flickr data set. Results suggest that such a control is indeed possible and the proposed method significantly outperforms other approaches that employs mainly degree-based mechanisms for controlling the information dissemination over the social graphs.

I. Introduction

Social networks have been used as one of the effective means of disseminating information about products for marketing and advertisement purposes [1], [2], [3]. Recent years have witnessed the emergence of vastly popular online social networks that aggregate millions of users and various interest groups together. Some of these networks such as Facebook also released APIs for application developers, content providers, and service providers to launch and spread new applications, contents, and services through the online social network platform. The impact has been that application developers, content providers, and service providers have had access to a large social network with millions of users to popularize their applications, contents, and services that are hosted outside the infrastructure of the social network provider. As online social networks are becoming a more common place for attracting users to consume applications, services, and contents, they start to have a substantial influence on the utilization of CPU cycles, cache spaces, and network bandwidth provided by the infrastructure owners [4]. Our work advocates that infrastructure providers for cloud services such as network operators and data centers must interface with online social network providers more closely to optimize their own resource utilization by controlling the viral adoption and consumption of applications and contents hosted on their sites. To this end, the network or data center operator should understand how information is spread over the social networks and how/whether this spread translates into an actual demand over time for different applications and contents.

When empirical data from various sources are closely examined, two significant patterns emerge in the adoption of products and services: (i) an independent pattern where the adopters do not have a previously established direct social relation with the earlier adopters, but learn about the product/service through other channels (e.g., advertisement broadcasts) and (ii) a viral pattern where the product/service is adopted contagiously between adopters through their social relations. The first pattern leads to a more regular access pattern converging to a steady state behavior. The second pattern on the other hand creates a burst of activity creating flash crowds in relatively shorter time spans. This latter pattern is the hardest to handle for resource provisioning and most often requires over-provisioning to serve potential peak demands. The viral pattern also affects security, as untrusted, even malicious, applications might propagate out of the control of the operator in the network. Thus, regularizing the demand due to such patterns is of paramount importance to have a more secure, a more even and a higher utilization of the infrastructure resources. Our work addresses this issue and provides methods for an infrastructure provider to social engineer and regularize the demand by exerting control over the hosted contents/applications. Without loss of generality, we will limit ourselves to content distribution networks as the infrastructure provider.

Consider the picture depicted in Fig. 1. This graph representation shows a portion of a social graph generated by a given social network. The nodes correspond to users and undirectional links (they can be also directional) indicate that the actions taken by one node are visible to the other nodes unless they are explicitly filtered. The filtering is typically done by the user who takes the action. In our scenario, however,

978-1-61284-231-8/11/$26.00 ©2011 IEEE

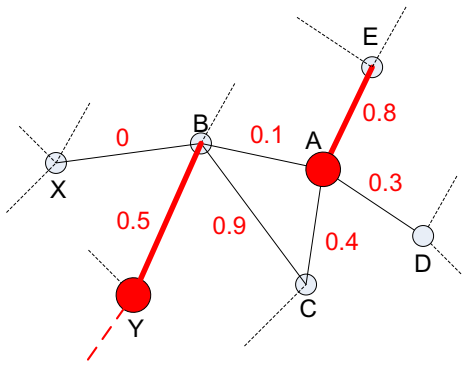

Fig. 1. Controlling the exposure of service/content adoption over the influence graphs.

we also let the content provider to temporarily or permanently filter the actions taken over their hosted content. We assume that the user's filtering rules are quasi-stationary and do not wildly change over time, while the content providers set their filtering rules dynamically. The social channel that makes the actions visible can take different forms such as wall posts, invitations, status updates, etc. The weights over the edges indicate the likelihood that a child node repeats the same action (e.g., download a particular video file) its parent node takes once the parent's node action becomes visible to itself. In Fig. 1, Nodes A and Y have already consumed a content. The thicker red colored edges indicate that their actions have not been filtered by the content provider and they have a chance to influence their neighbor at the other end of the edge. For instance, E sees that A consumed a given content, whereas B, C, and D cannot see A's action until the content provider removes the filter. Node B, however, sees that Y consumed the same content. We provide the formal model of influence graphs with directional edges in Section III.

In our proposed architecture, social network provider determines the influence probabilities based on past action history and exposes an influence graph similar to Fig. 1 to the content provider. Content provider in return sets or removes filtering rules for different contents and decides when the actions taken over the hosted content are allowed to propagate over which links. This allows the content provider to control the spikes and bursts in traffic due to the propagation over the influence graph, and to smooth the demand for content over time. As the social interactions between users are continuously monitored by the social network provider through other channels, the influence graph is continuously updated so that the filtering rules for future content propoagation are determined accordingly. As one of the main contributions of this paper, we present an algorithm to reduce the time to expose an event originated by a given set of users to all users, while setting an upper bound on the number of concurrent activities related to the event. For instance, if one user shares a video file, our algorithm will ensure that no more than a certain number of users have access to the video file at a given time, while attempting to reduce the number of time steps to make the video file virally available to all interested users without impacting the usage over longer time periods. Furthermore, we apply the proposed algorithm for limiting propagation of untrusted applications in the network by ensuring that such applications are not exposed to users with the highest influence as soon as they are launched, while at the same time exposing the application to the sufficient number of users to be able to evaluate its nature (malicious or trusted). We evaluate the algorithm based upon the actual data set from Flickr and find that it significantly reduces the time to disseminate a piece of data upon the social graph conditioned on setting a limit on the number of concurrent users.

The paper is organized as follows: in the next section, we describe related work on the influence modeling and its impact on on-line social network infrastructure. In Section III, we describe our modeling framework and introduce our algorithm to compute the reward achieved by exposing each link based on the influence graph. In Section IV, we validate the algorithm, and show that it outperforms significantly some simple heuristic based on the degree of the nodes in the social graph. We offer concluding remarks in Section V.

II. Related Work

The majority of previous research studying social influence mainly used the dynamics of disease spread to study information diffusion over social networks [5], where cascades have been analyzed by sociologists interested in *diffusion of innovation* [6]. Although numerous studies addressed the problem of tracking the propagation patterns of topics and influence through network spaces [1], [2], [3], the main problem that attracted most of the attention is *influence maximization*; which is finding an effective, small set of influential users that are sufficient to reach out for the majority of users in the social network [7]. Domingos and Richardson [8], [9] were the first to propose a probabilistic model for the influence-based interactions in the social network, as well as a heuristic for identifying influential users.

There are two basic models considered in literature for the process of information diffusion and product adoption: *Threshold models* [10], where a user adopts a new idea / product based on the aggregate influence exerted upon her by the neighboring peers, and *Cascade models* [11], where each social contact of a user has a single chance of influencing her once the contact adopts the new idea, based on some influence probability p_e. More recently, Kempe et al. [12] introduced a generalized framework that is able to capture the properties of both the *linear threshold* and the *independent cascade* models, as well as providing approximation guarantees on their proposed generalized model.

In terms of social networks, a lot of work has attempted to characterize user interactions in Facebook [13], [14]. Wei et al [15] considered the propagation of applications between users, using data from a game provider. Focusing on the services offered over the cellular network, Szabó and Barabási [16] attempted to discriminate the usage patterns of four services (email, wap, chat, multicast IM) based upon the underlying social behavior. Sun et al. [17] studied the influence

over actual Facebook data based on the propagation of items from the News feed. They show that the degree of a user is not the best predictor in computing the propagation of a chain of items, underlying the need for a better characterization of the influence in between users.

Finally, some research works have started to study the impact of the dynamics in between users in a social network upon the infrastructure which provide the services. This is the work most closely related to ours, as we use the influence of users to smooth the demand for services and data. Pujol et al [18] showed how to cluster the users of a social network over servers based upon the social network's connectivity graph. Nazir et al [4] considered the network-level effects of the application-level user behavior based upon Facebook. This is a measurement study, and it offers some guidelines to application developers and OSNs, but those are very high level, and only point to the direction of deriving some practical algorithms to make the delivery of on-line social networking content more efficient. We now turn to formally describing such algorithm.

III. Model and Method

A. Model

We consider a set of users generating demand for a content distribution network (CDN) through dissemination information about specific contents they consumed over an online social network. We assume that CDN operates under capacity constraint with respect to how many users or how many concurrent content requests can be handled at any given time.

For modeling purpose, we consider there is only one type of content in the network. However, this can be generalized to multiple contents with no loss of generality. We assume a time slotted model, where each time slot corresponds to the time it takes for a user to access the content. Once a user has access the content, this information ("user u has accessed this file") is made available. However, the CDN can control who, amongst the users connected to u, can view this information at each time step, and delay it if it suits its traffic engineering purpose. For instance, the social network provider may want to reduce congestion at the server and improve the QoS for the users who are allowed to access the content at each time epoch.

To optimize to whom the social network expose the user's activity, the CDN uses the *influence* users exert over each other. The social network can expose links (i.e. for connected users u and v, it shows the activity of u to v, exposing the link (u, v)) or users (i.e. it can show the activity of user u to all its neighbors) for a given content. Which links or nodes to be exposed for that content are however controlled by the CDN. We are interested in the following problem:

Problem P1: *Which k links should the CDN expose at each time step t, to minimize the time to share the content to all users*.

We define our influence network as a directed graph $G(V, E)$, where V represents the set of individuals in the network. Each directed edge $e(u, v) \in E$ indicates the direction of influence between nodes u and v, and is annotated with a probability p_e indicating the likelihood of node v being activated as a result of node u being active.

The influence p_e is acquired on-line, by monitoring previous activities. In this paper, we assume that the training period has completed and that the p_e's are given to us.

Consider an inactive node v and the set of its currently active neighbors $N(v) = \{u : e(u, v) \in E, active(u) = true\}$. Assuming independence of influence probabilities, the joint influence probability of N on v can be defined as:

$$P_{N(v)}(v) = 1 - \prod_{u \in N} (1 - p_{e(u,v)})$$

As a new neighbor of w of v gets activated, the new joint influence probability on u can be written as

$$\begin{aligned} P_{N(v)\cup\{w\}}(v) &= 1 - (1 - p_{e(w,v)}) \prod_{u \in N(v)} (1 - p_{e(u,v)}) \\ &= 1 - (1 - p_{e(w,v)})(1 - P_{N(v)}(v)) \\ &= P_{N(v)}(v) + (1 - P_{N(v)}(v)) \times p_{e(w,v)} \end{aligned} \tag{1}$$

Therefore, the local value of exposing a new link $e(w, v)$ from an active node w on the immediate neighboring inactive node v can be defined as

$$\begin{aligned} r_{e(w,v)} &= P_{N(v)\cup\{w\}}(v) - P_{N(v)}(v) \\ &= (1 - P_{N(v)}(v)) \times p_{e(w,v)} \end{aligned} \tag{2}$$

Similarly, we can also define the local effect of activating a given node u as the resulting expected increase in the number of its active neighbors

$$\begin{aligned} r(u) &= \sum_{v \in \bar{N}(u)} P_{N(v)\cup\{u\}}(v) - \sum_{v \in \bar{N}(u)} P_{N(v)}(v) \\ &= \sum_{v \in \bar{N}(u)} (P_{N(v)}(v) + r_{e(u,v)}) - \sum_{v \in \bar{N}(u)} P_{N(v)}(v) \\ &= \sum_{v \in \bar{N}(u)} r_{e(u,v)} \end{aligned} \tag{3}$$

where $\bar{N}(u) = \{v : e(u, v) \in E, active(v) = false\}$ represents the set of all currently inactive neighbors of u.

By adding the local effect of the destination node from equation 3 to the value of the exposed link defined in equation 2, we can formulate our global reward for each newly exposed link $e(u, v)$ as

$$R_{e(u,v)} = (1 - P_N(v)) \times p_{e(u,v)} \times (1 + \sum_{w \in \bar{N}(v) \wedge w \notin T} R_{e(v,w)}) \tag{4}$$

As can be noted in equation 4, the update formula for the edge rewards is recursive, thus to prevent cycles we keep track of the current set of traversed nodes T. The recursion terminates when there are no more inactive neighbors of a given node, or when an edge points back to an already traversed node in the current path. Introducing these conditions

to break cycles results in having the edge rewards dependent on the initial source of diffusion. Despite the fact that this adds in an overhead of recomputing the edge rewards with each new cascade depending on the source node, it limits the analysis to a subgraph $G'(V', E')$ including only the set of nodes that are reachable from the initial source and their interconnecting edges. Also, by reusing local computations, initializing the link rewards can be realized in $O(E')$, which is also the complexity of the full algorithm complexity.

B. Algorithm

Based upon equation 4, we can now describe the algorithm to solve the problem P1.

In the initialization phase, all the nodes in the social network are inactive except for the source of the cascade. As a result, the initial reward for each link in the network is computed with respect to all the neighbors of the destination node, until a cycle is detected. As the cascade is initiated, in order to maximize the influence at each step, we expose the link with the highest reward value $e^*(u^*, v^*) = \arg\max_{e(u,v)} R_{e(u,v)}$: $active(u) = true$ from a currently active source node, and update the joint influence on the destination node v^* according to equation 1. As the activation probability of the destination node v^* is changed, we need to update the rewards of all the edges that depends on $P_N(v^*)$. By exploiting the recursive nature of our rewards formulation, the required update can be achieved by successively propagating the change in the node's probability $\Delta P_N(v^*)$ along the edge rewards on all the back paths from $v*$ to the cascade source, other than the current activation path. The details of the update process is described in algorithm 1.

After exposing a link, its reward drops instantaneously to zero, indicating the fact that there's no more additional value that can be gained from this edge. However, following the activation dynamics, no update is required for the edge rewards as all the nodes on the activation path leading to the current exposed edge must be already active. The full details of the whole process of link exposure to maximize the spread is given in algorithm 2.

Algorithm 1 Reward Update

Require: Input graph $G(V, E)$, cascade source S, and exposed edge $e(u, v)$

Set $\Delta P_N(v) = (1 - P_N(v)) \times p_{e(u,v)}$
for each incoming edge $e'(w, v)$ such that $w \neq u \wedge R_{e'} > 0$ **do**
  Set $R^{old} = R_{e'}$
  Set $R_{e'} = \frac{R^{old}}{1-P_N(v)} \times (1 - (P_N(v) + \Delta P_N(v)))$
  Set $\Delta R = R_{e'} - R^{old}$
  for all back paths $L : w \rightarrow S$ **do**
    Set $\Delta R' = \Delta R$
    repeat
      Set $e_x(x, y) = L(0)$
      Set $R^{old} = R_{e_x}$
      Set $R_{e_x} = R^{old} + (1 - P_N(y)) \times p_{e_x(x,y)} \times \Delta R'$
      Set $\Delta R' = R_{e_x} - R^{old}$
      Remove $L(0)$
    until $L = \phi$
  end for
end for
Set activation probability $P_N(v)$ to $P_N(v) + \Delta P_N(v)$

Algorithm 2 Full Algorithm

Require: Input graph $G(V, E)$, cascade source S

Initialize all node activation probabilities P_N to zero
Set $P_N(S) = 1$
Recursively compute edge rewards starting from node S as defined in equation 4
Set $legitEdges = \{e(S, v) : e \in E\}$
while $legitEdges \neq \phi$ **do**
  Set $e'(u, v) = \arg\max_e R_e : e \in legitEdges$
  Set $R_{e'(u,v)} = 0$
  Update Rewards from $e'(u, v)$ as in algorithm 1
  Set $legitEdges = legitEdges \backslash \{e'(u, v)\} \cup \{e(v, w) : e \in E \wedge active(v) = true\}$
end while

IV. Experiments and Results

In order to evaluate our rewards algorithm, we performed a set of experiments on Flickr, a photo-sharing website based on a social network with friendships and family links. The dataset was sampled from a larger set provided by Alan Mislove [19]. Our sample consists of 7937 social links between 3615 different individuals. The adoption probabilities were computed using diffusion information for 2008 social groups over 104 days in Flickr. For the purpose of simulation, we used the general threshold model [12], which is based on setting a threshold value θ for each node, where the node is activated when the joint influence on it exceeds its specified threshold.

In the first set of experiments, we compare the number of nodes that could be activated by exposing a certain percentage of links in the network using our rewards algorithm, and compare the results to three different baselines: *random*, *degree* and *Greedy-PI*. The *random* baseline is based upon random choice of edges, while the *degree* baseline ranks the edges according to the degree of the destination node, following one of the well known heuristics to identify influential nodes. The final algorithm *Greedy-PI* represent the deterministic scenario where the actual node thresholds are observed and a greedy choice is made at each step to maximize the number of activated nodes.

As can be noted in figure 2, for a given percentage of exposed edges in the network, our algorithm achieves a higher percentage of active nodes compared to both *random* and *degree* baselines. On average, our algorithm is able to achieve ∼50% increase in the number of nodes that could be activated using the same percentage of exposed edges. Our proposed reward algorithm is also much closer to the *Greedy-PI* deterministic method that uses the perfect information about node thresholds than any of the other baselines.

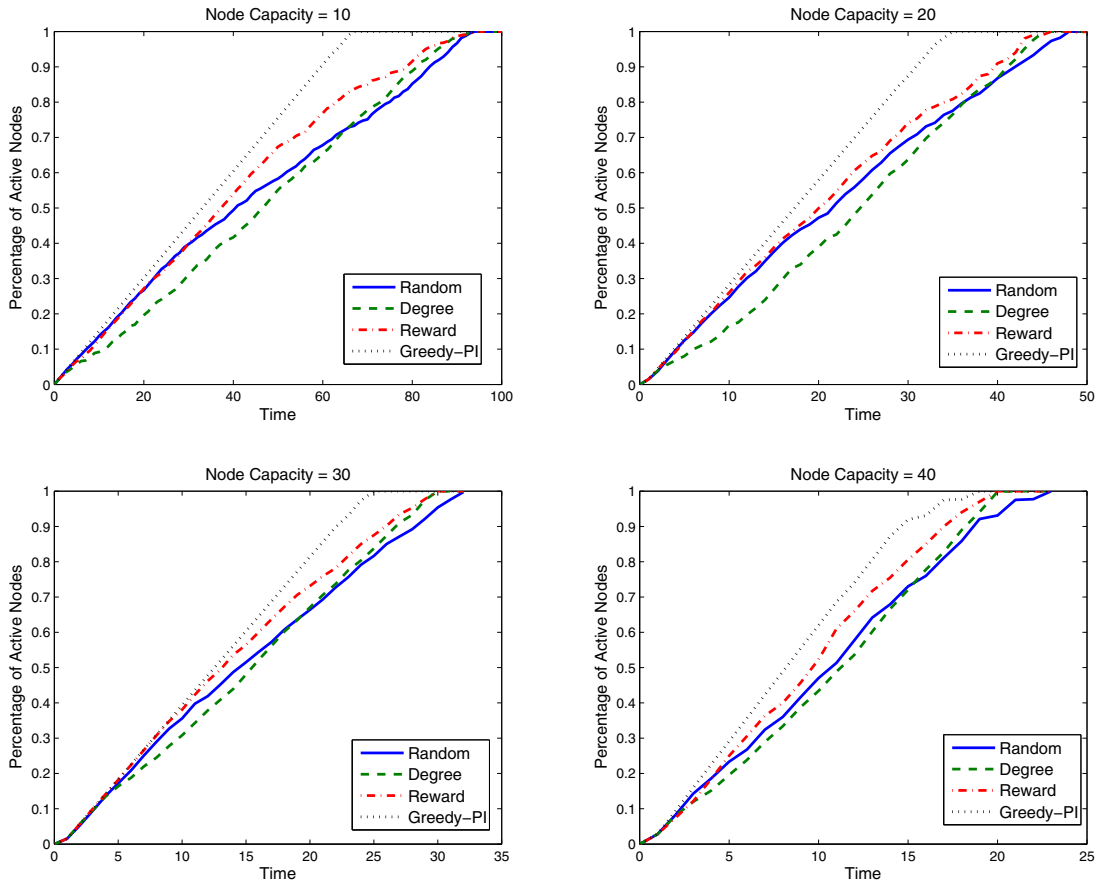

Fig. 3. Percentage of active nodes over time with limited node capacity

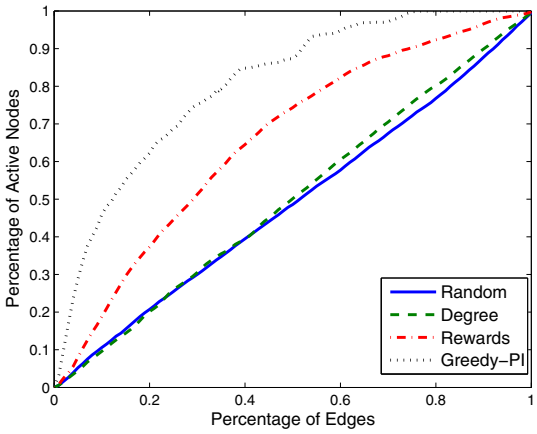

Fig. 2. Percentage of active nodes vs. exposed edges

In the second set of experiments, we set a target utilization in terms of number of nodes that need to be activated to meet our resource scheduling at each time point, which can be determined according to the current state of the network. In this set of experiments, at each time point, each algorithm chooses a set of edges to be exposed such that the resulting expected number of activated nodes according to the joint influence probability is equal to the target utilization. Note that in this set of experiments, the actual nodes that are activated are fed back to the algorithms after each time point, which makes the algorithms perform in a batch/offline mode in contrast to the online mode in the previous set.

As can be observed in figure 3, we note that our proposed rewards algorithm still outperforms the other two baselines for different values of target utilization. We also note that the performance of our algorithm is best when the the target utilization is minimal, which is attributed to the fact of having faster feedback to the algorithm on lower utilization rates, which enables our proposed rewards algorithm to be self adjusted sooner.

Our final set of experiments is related to the propagation of unverified applications or services in the network. In order to minimize the impact of potentially malicious applications on the network, we slow down the propagation of newly introduced services/applications over the network until they are analyzed and verified as legitimate. At the same time, we aim to achieve minimal impact on the future growth of the analyzed service if it turns out to be legitimate. In order to achieve that, we assume that an application is confirmed to be either malicious or legitimate after $t_x\%$ of time and we block $d\%$ of highest reward edges (set their reward to

0) in the network during that time period, therefore making application 'invisible' for a subset of high influence users until the application is verified. For that purpose, we introduce a damping factor, which represents the percentage of edges with highest rewards that are blocked in order to achieve the required propagation dampening. Figure 4 shows the results

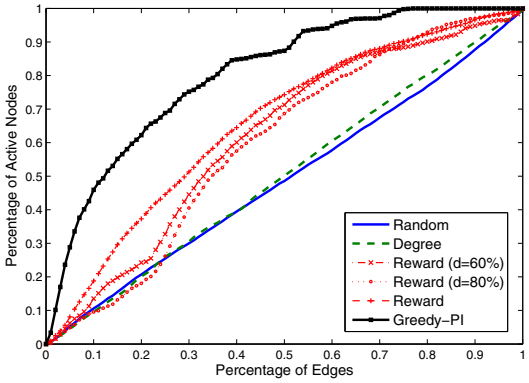

Fig. 4. Percentage of active nodes vs. exposed edges with reward dampening

of this set of experiments, where the propagation is dampened from 0% until 25% on the time scale. As can be noted in the figure, by increasing the damping factor, we are able to slow down application propagation using the proposed rewards algorithm. We also observe that, as soon as the block is removed (i.e. when the application is verified), the growth rate increases again to reach the same rate which would have been attained without having the initial blocking period. Thus, our proposed *rewards* algorithm can be efficiently used for controlling suspicious services without impacting future application adoption in the network.

V. Conclusion and Future Work

We have proposed a mechanism to use the influence in between users in a social network to perform shaping of the users' demand for content, so as to satisfy the CDN infrastructure or security constraints. We believe such mechanism has the potential for wide application in social networks, as flash demand for content could overwhelm the infrastructure. Our algorithm significantly improves the time to deliver the content under a number of concurrent download constraint, when compared with degree-based and random mechanisms.

These promising results lead us to some future investigations. Typically, the influence probabilities between different users in the social network change over time, subject to different interactions and cascades that occur over the social graph. For future work, we are planning on incorporating this temporal factor in our analysis, adjusting our reward algorithm to take these temporal changes into account in order to achieve an optimal resource usage policy without affecting the service adoption over the network.

The potential impact of untrusted applications on network performance remains an important issue for network operators. We examined a method for slowing down propagation of such applications and reducing their impact on network performance if they are proven to be malicious. However, a number of issues still remain open, and we intend to examine how adoption of applications by low influence users affects detection time (i.e. do detection mechanisms need more time to detect malicious applications if they are first adopted by low influence users) and consequently whether it is possible to detect malicious applications before the critical mass of infected users needed to bring down the network is reached. Hence, a more detailed analysis of impact of dampening factor on detection of malicious applications is needed.

References

[1] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information diffusion through blogspace," in *Proceedings of the 13th International World Wide Web Conference (WWW 2004)*, 2004, pp. 107–117.

[2] E. Adar and L. Adamic, "Tracking information epidemics in blogspace," in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2005)*, 2005, pp. 207–214.

[3] J. Leskovec, L. Adamic, and B. Huberman, "The dynamics of viral marketing," in *Proceedings of the 7th ACM Conference on Electronic Commerce (EC-2006)*, 2006, pp. 228–237.

[4] A. Nazir, S. Raza, D. Gupta, C. N. Chuah, and B. Krishnamurthy, "Network level footprints of facebook applications," in *ACM SIGCOMM conference on Internet measurement conference IMC'09*, New York, NY, USA, 2009, pp. 63–75.

[5] N. Baily, *The Mathematical Theory of Infectious Diseases and its Applications*, 2nd ed. Griffin, Lodon, 1975.

[6] E. Rogers, *Diffusion of Innovations*. Free Press, 1995.

[7] M. Gladwell, *The Tipping Point: How Little Things Can Make a Big Difference*. Back Bay Books, January 2002.

[8] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proceedings of the seventh ACM international conference on Knowledge Discovery and Data Mining (KDD'01)*, 2001.

[9] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *KDD'02, the eighth ACM international conference on Knowledge Discovery and Data Mining*, 2002.

[10] M. Granovetter, "Threshold models of collective behavior," *American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1987.

[11] J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, no. 3, pp. 211–223, 2001.

[12] D. Kempe, J. M. Kleinberg, and E. Tadros, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM international conference on Knowledge Discovery and Data Mining (KDD'03)*, 2003.

[13] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *WOSN '09, 2nd ACM workshop on Online Social Networks*, New York, NY, USA, 2009, pp. 37–42.

[14] A. Nazir, S. Raza, and C. N. Chuah, "Unveiling facebook: a measurement study of social network based applications," in *ACM SIGCOMM conference on Internet measurement conference IMC'08*, 2008, pp. 43–56.

[15] X. Wei, J. Yang, L. A. Adamic, R. M. de Araújo, and M. Rekhi, "Diffusion dynamics of games on online social networks," in *Workshop on On-line Social Networks, WOSN'10*, June 2010.

[16] G. Szabo and A. L. Barabasi, "Network effects in service usage," Nov 2006. [Online]. Available: http://arxiv.org/abs/physics/0611177

[17] E. Sun, I. Rosenn, C. Marlow, and T. Lento, "Gesundheit! modeling contagion through facebook news feed," in *International AAAI Conference on Weblogs and Social Media, ICWSM'09*, 2009.

[18] J. M. Pujol, V. Erramilli, G. Siganos, X. Yang, N. Laoutaris, P. Chhabra, and P. Rodriguez, "The little engine(s) that could: scaling online social networks," in *ACM SIGCOMM '10*, September 2010, pp. 375–386.

[19] M. Cha, A. Mislove, and K. P. Gummadi, "A Measurement-driven Analysis of Information Propagation in the Flickr Social Network," in *Proceedings of the 18th Annual World Wide Web Conference (WWW'09)*, Madrid, Spain, 2009.