ISSN 2319 - 5975

Volume 5, No.1, December 2015 – January 2016

International Journal of Networks and Systems

Available Online at http://www.warse.org/IJNS/static/pdf/file/ijns01512016.pdf

Enhancing Reliability in P2P Networks Using Social Capital Principles

Henry Hexmoor and **Nagalakshmi Satyanarayanan**

Computer Science Department

Southern Illinois University Carbondale, IL - 62901

ABSTRACT

Online file sharing is a commonly important day to day activity that is on the rise with proliferation of social media that relies ona reliable and trustworthy network. In a distributed networking environment such as a Peer-to-Peer network there are millions of users sharing files at each and every passing moment. Social capital is a quantity that reflects multiple attributes among nodes of a network such as power, relations, and trust as the most prominent element. The network is inhabited by various types of users and we are aspiring to making it more secure in terms of privacy and reduction of malware by using certain principles inspired from recommender systems and social capital.

1. INTRODUCTION

The goal of this paper is to study the drawbacks of peer to peer network paradigms and to propose a system where we use social capital as a set of evaluation criteria to make the network more reliable and resilient to malware. To accomplish this we use the principles of social capital, Peer to Peer paradigm of bit torrent, and recommendation systems.

Social capital is the expected collective or individual economic benefits derived from the preferential treatment and cooperation between individuals and groups in the network [6]. Although different social sciences emphasize different aspects of social capital, they tend to share the core idea that social networks have social value[9].

By and large, social network isa theoretical construct that is useful in the social sciences as a tool to study relationships between individuals, groups, organizations as well as entire societies. The term is used to describe a social structure determined by social interactions[14]. Social interactions plays an important role for the development of social capital in any dynamic network. Albeit, these interactions may not be secure from misuse by self-interested agents. To counter that, each agent must be capable of identifying reliable interaction allies for and by itself. Social capital involves certain aspects of social structures, mostly social networks which facilitate the members' actions inside the social structure or networks[6]. The existing methods for finding influencers primarily use the process of information diffusion to discover the nodes with maximum information spread. These models are limited to capturing the process of information diffusion and not the actual social value of collaborations in the network. Recently, a method has been proposed for finding influencers using the idea that people generate more value for their work by collaborating with peers of high influence [8]. The social value generated through such collaborations denotes the notion of individual social capital. They have also hypothesized and argued that players with high social capital are often key influencers in the network. They have proposed a value-allocation model to compute the social capital and allocate the fair share of this capital to each individual involved in the collaboration. We extend this work by using this concept to allocate leaders in this proposed model [8].

Peer-to-peer (P2P) computing or networking is a distributed application architecture that partitions tasks and work loads among a number of peers. Peers are equally privileged, equipotent participants in the application. They are said to form a peer-to-peer network of nodes. Peers make a portion of their resources, such as processing power, disk storage, or network bandwidth, directly available to other network participants, without the need for central coordination from servers or other stable hosts[12].Peers are both suppliers and consumers of resources, which is in contrast to the traditional client-server model in which the consumption and supply of resources are divided.

Although peer to peer networks are one of the most scalable and resilient strategies against malware and attacks by the mere facts of distributed storage and search, they are still under the threat of malware and may often contain files that are irrelevant. In most cases the peer to peer networks involves data transfer from one user to another without using an intermediate server. Corporations that are developing P2P applications have been involved in numerous legal battles, primarily in the United States, over conflicts with copyright laws[5].

We adopted *recommendation systems* in our approach to observe a user's behavior pattern online. These are a subclass of information filtering system that seek to predict the 'rating' or 'preference' that a user would give to an item. [4].

Recommendation rating can be achieved using collaborative and content based recommendation systems also known as the User-User & Item-Item approaches. Recommender systems have become extremely common in recent years and are applied in a variety of applications. The most popular ones are probably the movies, music, news, books, research articles, search queries, social tags, and products in general. However, there are also recommender systems for intangibles such as experts, jokes, restaurants, financial services, life insurance, persons (i.e., as in online dating), and *twitter* followers [4].

Collaborative filtering methods are founded on collecting and analyzing a large amount of information on users' behaviors, activities, or preferences and predicting what users will like based on their similarity to other users. A key advantage of the collaborative filtering approach is that it does not rely on machine analyzable content and therefore it is capable of accurately recommending complex items such as movies without requiring an understanding details of the item itself. Collaborative filtering is based on the assumption that people who have agreed in the past will agree in the future, and that they would like similar kinds of items as they liked in the past.

When building a model from a user's profile, a distinction is often made between explicit and implicit forms of data collection. Methods for explicit data collection include asking a user to rate an item on a sliding scale, asking a user to search, asking a user to rank a collection of items from the most favorite to least favorite, and presenting two items to a user and asking her to choose the preferred one of them, and asking a user to create a list of items that she likes.

In contrast, ways of collecting implicit data are observing the items that a user views in an online store, analyzing item/user viewing times[10], keeping a record of the items that a user purchases online, obtaining a list of items that a user has listened to or watched on his/her computer, and analyzing the user's social network and discovering similar likes and dislikes.

Content-based filtering methods are based on a description of the item and a profile of the user's preference [11]. In a content-based recommender system, keywords are used to describe the items. Additionally, a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are similar to those that a user liked in the past. In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended. This approach has its roots in information retrieval and information filtering research.

One of the most common peer to peer systems is known as *bit torrent* (Schulze and Mochalski, 2009), which described next.Bit torrent as a peer to peer based file sharing system where users on the network share files, music, movies, etc. where each file is divided into many parts and is distributed on a network. A user who uploads the first file is known as the *seeder*. A user is required to reserve a specific part of her bandwidth for uploading files in order to be able to download files. But this strategy suffers from the drawbacks of unreliability due the presence of malicious files. To send or receive files the user must have a BitTorrent client; a computer program that implements the BitTorrent protocol. Some popularclients
include Xunlei, Transmission, µTorrent, Mediaet, Vu ze, and BitComet. BitTorrent trackers provide a list of files available for transfer, and assist in transferring and reconstructing the files. The best-known BitTorrent tracker is the Pirate Bay. As of January 2012, BitTorrent was utilized by 150 million active users (according to BitTorrent, Inc.). Based on this figure, the total number of monthly BitTorrent users can be estimated at more than a quarter of a billion (bittorrent.com, 2012). The BitTorrent protocol can be used to reduce the server and network impact of distributing large files. Rather than downloading a file from a single source server, the BitTorrent protocol allows users to join a "swarm" of hosts to upload to/download from each other simultaneously. Using the BitTorrent protocol, several basic computers, such as home computers, can replace large servers while efficiently distributing

files to many recipients. A seeder is one who uploads the file and a *leecher* that is a user who downloads the file without making any contribution.

A user who wants to upload a file first creates a small torrent descriptor file that they distribute by conventional means (i.e., web, email, etc.). They then make the file itself available through a BitTorrent node acting as a seed. Those with the torrent descriptor file can give it to their own BitTorrent nodes, which—acting as peers or leechers—download it by connecting to the seed and/or other peers.

The file being distributed is divided into segments called pieces. As each peer receives a new piece of the file it becomes a source (of that piece) for other peers, relieving the original seed from having to send that piece to every computer or user wishing a copy. With BitTorrent, the task of distributing the file is shared by those who want it; it is entirely possible for the seed to send only a single copy of the file itself and eventually distribute to an unlimited number of peers.

Through *flooding mechanism*, although the likelihood of finding the required files increases, it also promulgates various issues regarding which, several studies on BitTorrent have indicated that a large portion of files available for download via BitTorrent contain malware. In particular, one small sample [2] indicated that 18% of all executable programs available for download contained malware. Another study [17],has claimed that as much as 14.5% of BitTorrent downloads contain zero-day malware, and that BitTorrent was used as the distribution mechanism for 47% of all zero-day malware they have found.

Corrupted data can also be distributed on P2P networks by modifying files that are already being shared on the network. For example, on the FastTrack network, the registered investment advisory RIAA managed to introduce faked chunks into downloads and downloaded files that were mostly MP3 files. Files infected with the RIAA virus were unusable afterwards and contained malicious code. The RIAA is also known to have uploaded fake music and movies to P2P networks in order to deter illegal file sharing[15].

Since each node plays a role in routing traffic through the network, malicious users can perform a variety of routing attacks, or denial of service attacks. Examples of common routing attacks include "incorrect lookup routing" whereby malicious nodes deliberately forward requests incorrectly or return false results, incorrect routing updates where malicious nodes corrupt the routing tables of neighboring nodes by sending them false information, and incorrect routing network partition where when new nodes are joining they bootstrap via a malicious node, which places the new node in a partition of the network that is populated by other malicious nodes[18].

2. APPROACH

The main idea is to use social capital as a set of evaluation criteria in order to enhance the reliability of a peer to peer paradigm, say bit torrent. Every user needs to reserve a portion of her bandwidth for uploading in a peer to peer network in order to be able to download files. But a common phenomenon observed among selfish agents on the network is uploading irrelevant files with misleading names and sharing that in order to download files. But bit torrent suffers the problem of copyrighted and malicious files being published on the internet. Most organizations use the concept of torrent poisoning, which is intentionally sharing corrupt data or data with misleading file names using the Bit Torrent protocol. This practice of uploading fake torrents is sometimes carried out by anti-piracy organizations as an attempt to prevent the sharing of copyrighted content among peer and to phish for the IP addresses of downloaders. But this leads to a scenario where a user might lose all her downloaded data irrespective of if it was purchased or downloaded illegally. The Nopir-B worm, which originated in France, poses as a DVD copying program and deletes all the mp3 files on a user's computer, regardless of whether or not they were legally obtained.

In this paper we propose a more reliable form of file sharing by using the concept of recommendation systems using both content based and collaborative filtering and divide the users among a large network in clusters. The division criteria is based on similar interests shared by a group of users. The common behavior among agents is that agents with similar interests tend to bond closely thus the value of social capital between them is greater as compared to agents with varied interests. Thus a group of users who like the genre comedy are grouped into clusters and those who like different genres such as horror, romance, thriller, action are grouped into their own respective clusters where the social distance between them is one. The users get to rate each other depending upon their personal experiences with the other users. Suppose two users in a cluster share files, a reliable agent would send non malicious and uncorrupted files whereas a selfish agent would share hazardous content. Depending on the files received the recipient rates the sender.

Using a recommendation system helps us observe the pattern of usage of a user's interests and one can easily determine the reliability of a user on the internet based on her history browsing patterns, type on content posted by the user and the user's behavior on the internet.

Once a user's pattern of browsing is observed, she is recommended to join a cluster containing a group of users sharing similar interests where the user with maximum interactions and maximum positive rating from majority of the recipients is chosen as the cluster head.

The leader can be viewed as the gate keeper for the domain of information whom the agents in the cluster consult so as to know the reliability of new peers entering the network. This is a dynamically changing paradigm where the leaders keep changing and information is updated with every new user joining the network.

Every user needs to risk atleast one interaction in order to know if the interactive users can be trusted or not. Thisrisk cannot be avoided but it reduces the possibilities of encountering malicious peers multiple times there by reducing it to a minimum of one to two.

If a user who has not been amalgamated into a social cluster network, that user is temporarily added to that network where she is showing interest or looking for at that particular instant.

Next, we introduce the incorporation of social capital in peer to peer sharing for added reliability.

Let us consider a typical bit torrent file sharing scenario where multiple users interested in multiple genres attempt to download movies. The goal is to increase reliability of the system and to avoid scenarios where the user receives malicious or missing files. Let us denote the set of users as U, genres as G, clusters as C. Frequency of social interactions is denoted with SI and interaction count with IC respectively. Social capital value is denoted with SV and social distance with SD.

The following algorithm summarizes our strategy.

1. **for** U = {u1,u2,u3,…} for each cluster {c1,c2,c3,…}∈C
2. For G = {g1,g2,g3,…}
3. Do run the recommendation system algorithm on U to create a set of different clusters C based on G.
4. **for** any cluster Cn,
5. Enable file sharing among u1,u2.
6. increase IC for every interaction, IC++
7. if interaction is positive increment SV by one, SV++
8. **else**
9. decrement it by one, SV--
10. **if** SI(un) >SI(u1,u2,…) where u1,u2,…un ∈Cn
11. and if SV(un) > SV(u1,u2..)
12. make un the leader of Cn
13. Leaders are updated periodically.
14. Leaders of c1,c2,c3, will be responsible for providing recommendations to other leaders of cn-1,cn,cn+1,…
15. The social distance between peers in a cluster is 1.
16. End

3. COMPUTER SIMULATIONS

We used Netlogo to show the simulations of our work. We used the following models to illustrate and study the effects on the network before and after applying our algorithm. Virus on a network developed and reported in[16]and [19] demonstrates the spread of a virus through a network. Although the model is somewhat abstract, one interpretation is that each node represents a computer, and we are modeling the progress of a computer virus (or worm) through this network. Each node may be in one of three states: susceptible, infected, or resistant. In the academic literature such a model is sometimes referred to as an SIR model for epidemics. Figure 1 is a representation of this model. The red nodes in the model represent infected nodes and the green nodes represent healthy reliable nodes.

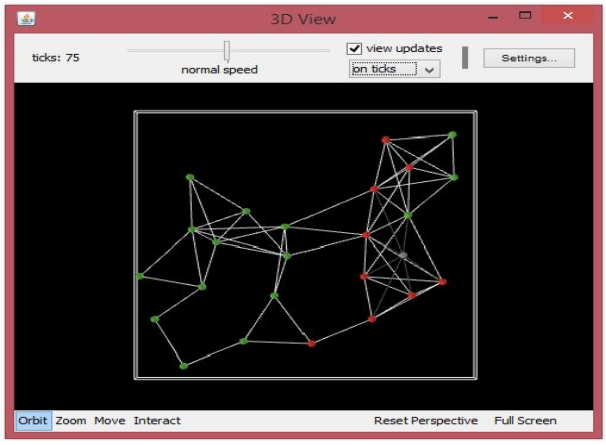

Figure 1. Virus on a network

We used the Team assembly model reported in[1]and [19] to illustrate how the network would look like after the application of our algorithm.But we make certain modifications to the assumptions made in the model for illustration purposes.

Figure 2 represents the team assembly model and the blue color depicts the good nodes separated into clusters, the red color depicts the bad and malicious nodes, and the yellow color depicts possibly infected nodes. As we can see from the first model, virus spreads rapidly on a network. But after the application of our model, we can observe that the spread of virus is relatively low.

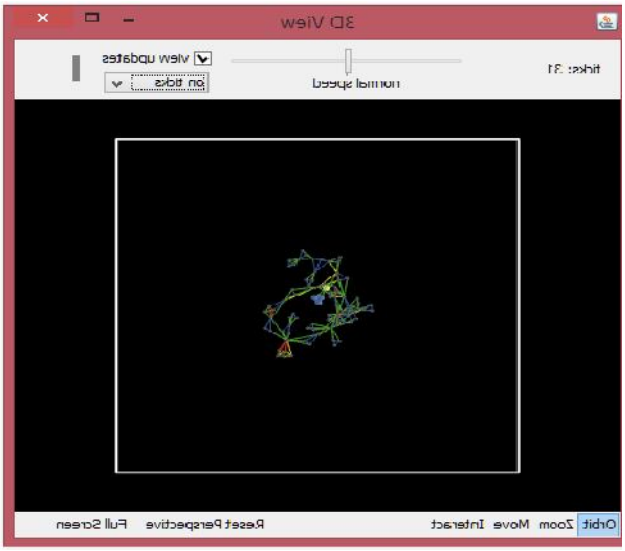

Figure 2. Team assembly

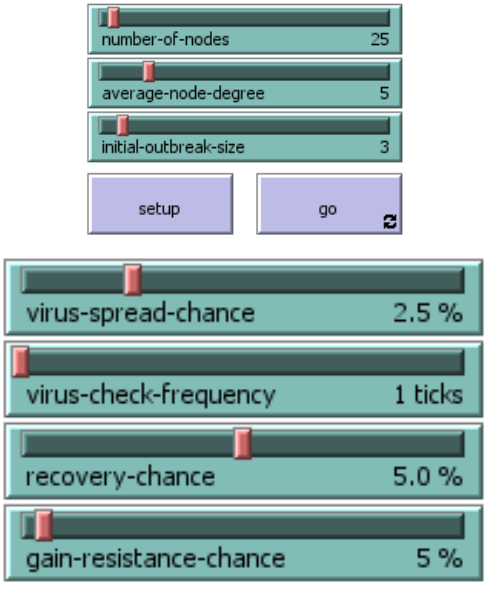

Figure 3. Virus on a network components

For representing the peer to peer model, we used the Virus on a network model, where we can set the number of nodes as per our requirement as shown in Figure 3. The average number of connections each node has is 5 and this can be changed as per the users need. With an initial number of 3 infected nodes and with the virus spread parameter as 2.5% which is very minimal, we can observe that the virus spreads rapidly, similar to peer to peer network.

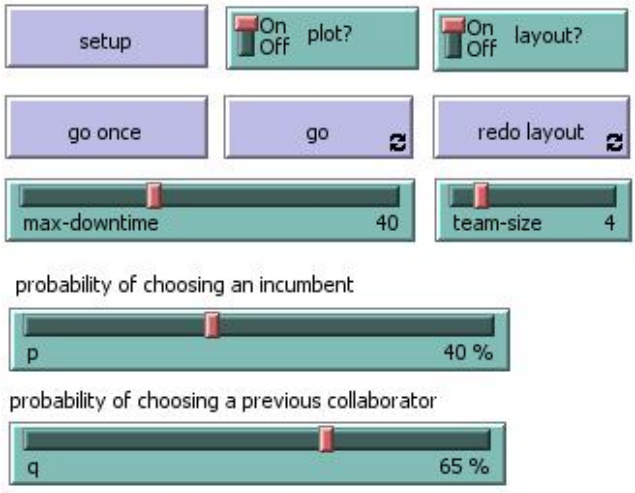

Figure 4. Clustered nodes components

The terms used in figures 4 are defined below:

- TEAM-SIZE: the number of agents in a newly assembled team.
- MAX-DOWNTIME: the number of steps an agent will remain in the world without collaborating before it retires.
- P: the probability an incumbent is chosen to become a member of a new team
- Q: the probability that the team being assembled will include a previous collaborator of an incumbent on the team, given that the team has at least one incumbent.

The *newcomer* represents an agent who has never collaborated, *component-size* represents current running size of component being explored. *Giant-component-size* is the size of largest connected component and *components* is a list of connected components.

The expression *incumbent?* returns true if an agent has collaborated before, *in-team?* returns true if an agent belongs to the new team being constructed. *Downtime* is the number of time

steps passed since the agent last collaborated and *explored?* is used to compute connected components in the graph. The attribute *new-collaboration?* Returns true if the link represents the first time two agents collaborated.

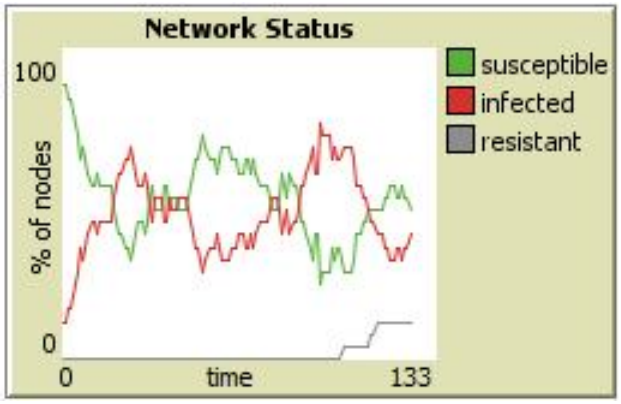

Figure 5. Depicting the network status graph

Figure 5 depicts the rate of virus infection in any network and the graph represents the states of nodes in the virus in the network model. It depicts that the resistance of the nodes eventually recedes and more nodes become susceptible to infection.

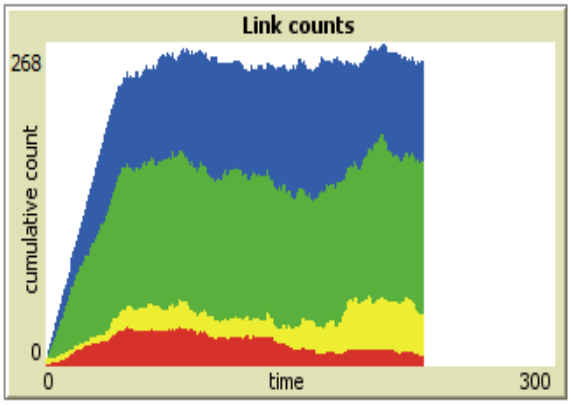

Figure 6. Depicting the link counts

The graph in Figure 6 represents the graphs obtained after the application of our algorithm, it depicts the degree of interaction in each cluster, i.e. it represents the interaction count. Interaction count is an increasing function which increases as the nodes begin to interact. The differences in the degree of interaction in each cluster varies and it's different for every iteration and it is a random effect as there could be different number of agents interacting in different clusters at any given point of time.

Figure 7. Depicting the % of agents in the giant component

The percentage *of agents in the giant component* represents the size of the largest connected component which is the leader node, in our model. The graph runs until the simulation is stopped, meaning end of one iteration of interaction. When an infected node enters the cluster, it shows a sudden increase in infection, but once that node is removed, it goes back to showing the lower infection rates. But in the assumptions that we make for our model, team assembly represents our approach and the virus in a network model represents infection rate in bit torrent, this graph shows the degree of infection which is always lower in our approach at any given time compared to that of bit torrent.

The graph in Figure 8 shows the results before and after the application of our algorithm.

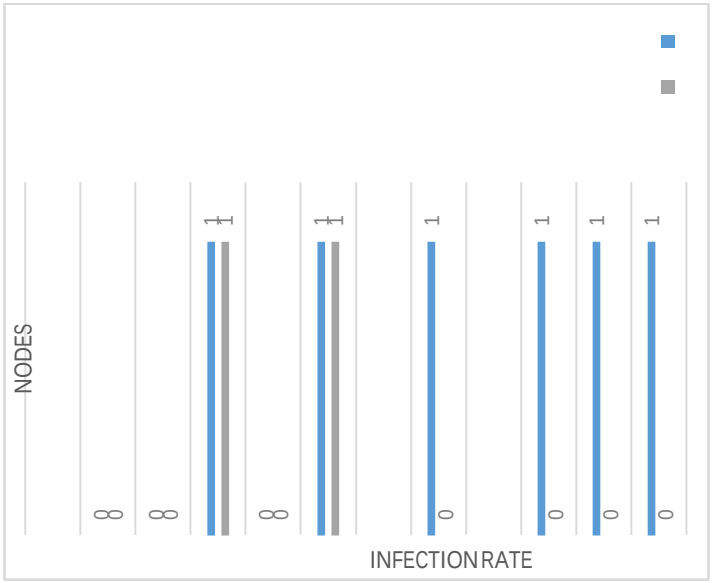

Figure 8. Graph depicting results

The blue lines show the infection rate in a typical peer to peer, bit torrent paradigm and the grey lines depict the results and lowered infection rate upon applying our approach. The graph was plotted keeping in mind a typical file sharing occurring inside a cluster i.e. after we ran the recommendation filters on the nodes in the network.

Considering five interacting nodes, sharing files among one another, among which one node is an infectious node. During the first phase, all the nodes interact with one another and the interaction count and the social value of each node is determined. As we can see in the graph, there is no infection is the first phase.

During the second phase the nodes with negative social value are removed from the network, thereby saving all the nodes in the cluster from getting infected. Thus as shown in the graph, results computed using peer to peer show that the entire cluster has been infected whereas in our approach, the infection is detected in the first phase and the infected node has been removed in the second phase thereby keeping the network secure from the third phase onwards.

Table 1 is a representation of interaction in a bit torrent system, on a typical simple cluster with five nodes and one infected node. We observe the contrast in degrees of infection in a peer to peer network and our approach. The node (here node 3) in red represents the infected node. The nodes 1,2,4,5 are not infected. When node 1 interacts with node 2 as shown in the first row, there is no infection. As node 2 interacts with, there is no infection. In phase one, node 3 is not interacting with any other node at this point. Node 4 interacts with 5 and there is no infection. However, node 5 interacts with node 3 and node 5 is infected.

In phase 2 node 1 is interacting with node 3 and it becomes infected.

In phase 3 node 1 interacts with node 2 and node 2 is now infected and node 2 interacts with node 4 and node 4 is also infected and node 4 and 5 which are already infected interact among each other, thereby leaving the whole network infected.

Peer-to-Peer		
Node	Interacting With	Infected?
Phase 1:		
1	2	N
2	4	N
3	Null	Y
4	5	N
5	3	Y
Phase 2:		
1	3	Y
Phase 3:		
1	2	Y
2	4	Y
4	5	Y

Table 1. Phases of interaction in a Peer-to-Peer Network

Our Algorithm								
Node	Interacting with	IC of Node	IC of Interacting node	SV of Node	SV of Interacting node	Infected?	Recovery	Removed
Phase 1:								
1	2	1	1	1	1	N		
2	4	2	1	2	1	N		
3	Null	0	Null	0	0	Y		
4	5	2	1	2	1	N		
5	3	2	1	2	-1	Y	5	
Phase 2:								
1	3	2	2	1	-2	N		3
Phase 3:								
1	2	3	3	2	3	N		
2	4	4	3	4	2	N		
4	5	4	2	3	3	N		

Table 2. Phases of interaction in our approach

In contrast to Table1, Table 2 is a representation of the phases of interaction in our implemented approach. Node 3 is the infected node in our approach. Node 1,2,4,5 are healthy nodes. In phase 1 node 1 interacts with node 2, so the IC of node 1 and node 2 is 1 and the SV of 1 and 2 is 1 as they are healthy nodes.

Then node 2 and 4 interact, IC of node 2 is 2 and node 4 is 1. SV of node 2 is 2 and SV of node 4 is 1. Node 3 is not interacting at this point of time. There is no infection.

Then node 4 and 5 interact, IC of node 4 is 2 and node 5 is 1. SV of node 4 is 2 and SV of node 5 is 1. There is no infection.

Now node 5 interacts with node 3 the IC of node 5 is 2 and node 3 is 1. The SV of node 5 is 2 and node 3 is -1 as 3 is an infected node. Node 5 is set to recovery.

In phase 2 node 1 is interacting with infected node 3 the IC of both nodes is 2. The SV of node 1 is 1 and

SV of node 3 is now -2 and now node 3 is set for removal from the cluster.

In phase 3 node 3 is removed and the nodes continue to interact with one another without infection. Thus, for any given number of n nodes at any point of time with x infected nodes, our approach would always yield a better result than the existing system.

4. CONCLUSIONS

Our main focus in this research has been on improving the existing systems that maintain reliability in the Peer to Peer paradigms mainly file sharing. We have proposed a model by using social capital principles and recommender systems that ensures a safer and more reliable file sharing in the networking environment that makes use of the different capitals that make up the social capital of an agent and by using various filtering systems to improve clustering in the network, thereby enabling us to have well defined roles for each agent in the network.

Also, to better handle requests we could use a proxy or a cache to redirect requests efficiently into the clusters. We could deploy a system similar to Hadoop Distributed File Systems, where a master node keeps track of all the nodes and incorporates node redundancy in case of a node failure.

Using distributed hash tables for node lookup and ratings are a good way to keep track of each node's social value. By using our approach after the very first interaction, it's possible to determine if a node is reliable or not. Due to early predictability of a node's trustworthiness, it is evident that a reliable node would not choose to interact with an unreliable node.

When a node moves from one cluster to another, its global trust value is taken into account to determine its reliability so that nodes do not have to be treated as a new node in each cluster, which saves the reliability evaluation time. However, if a peer chooses not to rate a peer with which it has interacted, then it would reduce the probability of accurate determination of social value. But its highly likely that a node which has had a bad experience would give a bad rating to make sure that the reputation of the node it has interacted with, goes down.

REFERENCES

1. Bakshy, E. and Wilensky, U., 2007. NetLogo Team Assembly model. http://ccl.northwestern.edu/netlogo/models/TeamAssembly. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
2. Berns, A.D., Jung, E., 2008.Searching for Malware in Bit Torrent, University of Iowa, via TechRepublic. Archived from the original on 1 May 2013. Retrieved 7 April.
3. BitTorrent and µTorrent Software Surpass 150 Million User Milestone,in Bittorrent.com. 9 January 2012.Archived from the original on 26 March 2014. Retrieved9 July 2012.
4. Ricci, F., Rokach, L., and Shapira, B.,,2011.Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, pp. 1-35
5. Glorioso, A.Ugo Pagallo, Ruffo, G., 2010. "The Social Impact of P2P Systems". In Shen et al. Handbook of Peer-to-Peer Networking. Springer.
6. Hexmoor, H., Alqithami, S., 2012. Social Capital in Virtual Organizations, Fourth International Conference on Intelligent Networking and Collaborative Systems, ACM press.
7. Helmstadter, E, 2003 The Institutional Economics of Knowledge Sharing: Basic Issues, In Helmstadter, E. (ed.) The Economics of Knowledge Sharing: A New Institutional Approach. Cheltenham & Northampton, M.A.: Edward Elgar, pp. 11-38.
8. Subbian, K., Sharma, D., Wen, Z., Srivastava, J., 2013. Finding Influencers in Networks using Social Capital, IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ACM press.
9. Putnam, R., 2000, "Bowling Alone: The Collapse and Revival of American Community" (Simon and Schuster).
10. Parsons, J., Ralph, P., Gallagher, K. 2004. "Using viewing time to infer user preference in recommender systems". AAAI Workshop in Semantic Web Personalization.
11. Brusilovsky, P., Kobsa, A., 2007. The Adaptive Web: Methods and Strategies of Web Personalization, Springer.
12. Schollmeier, R., 2002. A Definition of Peer-to-Peer Networking for the Classification of

Peer-to-Peer Architectures and Applications, Proceedings of the First International Conference on Peer-to-Peer Computing, IEEE.

13. Schulze, H., Mochalski, K., 2009. "Internet Study 2008/2009". Leipzig, Germany.
14. Scott, J. 2000. Social Network Analysis: A Handbook (2nd edition). Thousand Oaks, CA: Sage Publications.
15. Sorkin, A. R., 2003. "Software Bullet Is Sought to Kill Musical Piracy". New York Times.
16. Stonedahl, F. and Wilensky, U. 2008. NetLogo Virus on a Network model. http://ccl.northwestern.edu/netlogo/models/VirusonaNetwork. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
17. Vegge, H.., Halvorsen, M., Finn, M., Nergård, R., Walsø, R., 2009. Fools Download Where Angels Fear to Tread, Fourth International Conference on Internet Monitoring and Protection, IEEE press.
18. Vu, Q. H., Hieu, Q., and Chin, B.,2010. Peer-to-Peer Computing: Principles and Applications. Springer.
19. Wilensky, U., 1999. NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.