Mitigation of Malware Proliferation in P2P Networks using Double-Layer Dynamic Trust (DDT) Management Scheme

Lin Cai and Roberto Rojas-Cessa

***Abstract*—Peer-to-peer (P2P) networking is used by users with similar interests to exchange, contribute, or obtain files. This network model has been proven popular to exchange music, pictures, or software applications that are saved, and most likely executed at the downloading host. At the expense of this mechanism, worms, viruses, and intruding files find an open front door to the downloading host and giving them place to a very convenient environment for successful proliferation throughout the network. Although virus detection software are currently available, this countermeasure works in a reactive approach and most times isolated manner. In this paper, we consider a trust management scheme to contain the proliferation of viruses in P2P networks. Specifically, we propose a trust management system based on a two-layer approach to bound the proliferation of viruses. The new scheme is called Double-layer Dynamic Trust (DDT) management scheme. Our results show the proposed scheme bounds virus proliferation. With this approach, the number of infected hosts and proliferation rate are limited to small values. We compare our results to other existing approaches.**

***Index Terms*—Malware, P2P, peer-to-peer networks, trust management, virus proliferation.**

I. Introduction

Perhaps the simplest service model of a connection between two Internet hosts is the one used in peer-to-peer (P2P) networking, where a host can perform as a client or server. Current models in the Internet defines the provider of a service as a host, used almost exclusively as a server for technical and economical reasons, as being in charge to handle a large number of requests as a centralized entity.

P2P networks have the potential of converting any host into a data server and to use it as a part of a large system for disseminating information without the limitations of using a single (host) interface. This distribution potential is currently under the scrutiny for massive applications such as IPTV [1], [2], where video sources can rely on intermediate peers for further distribution of content. Furthermore, any user with Internet access with an acceptable bandwidth can participate in complex distribution networks, as proven by Napster [3] and Gnutella [4] for sharing music files.

A peer user, usually interested in the content of the received information, pre-approves storing the downloaded file and, most likely, executes it. This pre-acceptance process leaves a front door for viruses to the local host. Furthermore, other users can be encouraged to download popular Internet files for exploration, therefore creating an incubating environment for viruses.

Several interesting studies about virus proliferation have been presented [5]-[7]. They consider a network topology and features that describe the proliferation profile of a specific virus. Among other properties, viruses tend to have a spreading rate in function to the network density. Analysis of virus proliferation models is beyond the scope of this paper. Viruses or malware[1] have usually a specific destructive objective, whether they aim to the host computer, to retrieve user information that can be illegally profitable, or to affect communication resources (e.g. denial of service). Depending on the characteristics, viruses in a host may or may not affect other stored files.

The general countermeasure in a host against viruses is the use of an anti-virus program, which task can be coarsely divided into detecting a computing threat and removing it from the host. The successful detection by this protection software is based on the knowledge of existing hazardous files or software and their properties or signature for identification. Therefore, a new virus can be unnoticeable hosted in a peer until the detection program is updated for its identification. During this detection delay, the virus could be downloaded by another peer and so on. Furthermore, after a virus is detected in a peer, the detection software may remove the threat. However, this information might be kept from other peers as it may be considered information of only local significance.

Trust management schemes aim to distribute information about peers in different networks scenarios to decrease the degree of effect of misbehaving hosts [8]-[10]. In [11], a dynamic trust management scheme is proposed. This scheme is based on localized trust evaluation and in alert dissemination to prevent others from downloading a file from a suspicious peer. The scheme aims to limit the proliferation of malware under the assumption that there is no local file infection. In other words, when a virus-free peer downloads a file containing viruses, other existing files in the peer are not infected. However, viruses not only could specifically attempt to spread themselves but also infect the other files within the P2P network or pursue further hardware and software damage at the host of network level. Although the authors didn't assign

This work is supported in part by National Science Foundation under Grant Award 0435250.

Lin Cai and Roberto Rojas-Cessa are with the Networking Research Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA. Email: {lc76, rojas}@njit.edu.

[1]We may refer to virus or malware interchangeably in this paper as we are targeting those with similar characteristics in their proliferation.

a name to the scheme in the paper, we call this scheme Dynamic Threshold Management (DTM) in the remainder of this paper for the sake of brevity.

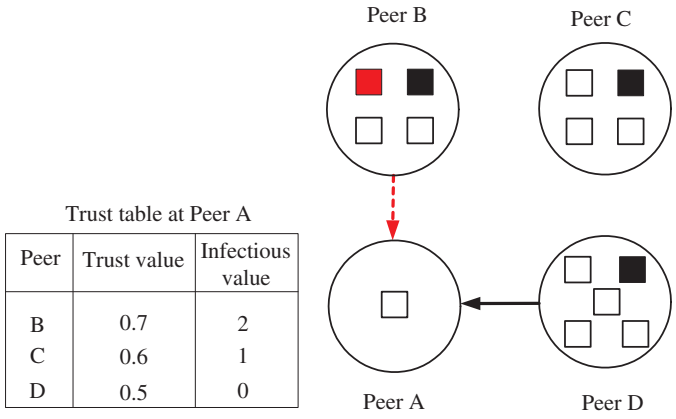
In this paper, we discuss the performance of the current P2P trust management strategy with consideration of internal file infection and show that file infection has the potential to underscore proliferation countermeasures. To bound virus proliferation, we propose the Double-layer Dynamic Trust (DDT) management scheme, which uses a two-layer trusting strategy aimed to alleviate the impact of the internal infection. The results show that our scheme trusting is efficient for infection control in P2P systems. We also analyze the influence of the propagation delay on the system performance, and observe how delayed alerts benefit network infection as informed peers cannot prevent clean peers from downloading files from infected peers in a timely fashion.

The remainder of this paper is organized as follows. Section II describes the proposed scheme based on dynamic trust management, the terms and the parameters for evaluation of peer trust, and the operation of the proposed management scheme in a P2P network. Section III shows the performance results obtained through computer simulation. Section IV presents our conclusions.

II. DDT Scheme

In this paper, we propose the DDT management scheme to bound malware proliferation in a P2P network. In this two-layer approach, each peer has a trust table that keeps two trust values. The first parameter, similar to the one used in DTM [11], is a trust value corresponding to the other peers. A trust value at peer A about peer B means the probability A has that a contaminated file would be downloaded from B. The higher the trust A has on B, the smaller the probability. Any peer in the system that is trusted by any other peer is called trustee and any peer that trusts a trustee is called truster. The second trust value in this table is designed to prevent internal infection, which is defined as the action of an infected file or malware that infects other files in a host by using this host as a means of local proliferation. This trust value is called the infectious value. The infectious value at peer A about peer B indicates the possibility of internal infection from downloading a file from peer B. The larger the infectious value is, the higher the possibility of an internal infection from an infected file downloaded from B. This means that A would consider less likely to perform a download from B.

The following is an example of how the proposed algorithm works. Let us consider that peer A wants a file and there are three possible trustees B, C, and D who have the desired file. Figure 1 shows a pictorial description of this example. The black square in the figure represents the requested file, the red square in peer B represents an infected file, which has infected the other files in peer B with probability P_I. In the DTM scheme, peer A chooses the peer that has the highest trust value at A. Peer A then chooses peer B as the downloading source. In our proposed scheme, the higher the infectious value is, the more severe the possibility that an infection has occurred in the corresponding peer. Peer A then chooses one with the lowest infectious value from its possible trustees. In this example, peer A selects peer D as the downloading source since its infectious value for D is 0, which is the lowest among B, C, and D. In this way, the system guarantees that a peer perform a download from the possibly cleanest source. Different from another schemes, we consider that a file can be infected by another file stored at the same peer. For example, as this figure shows, peer B has an infected file which is different from the one requested by A. Therefore, if peer A had selected the desired file from peer B, this file may have been infected (with probability P_I) and all the files at peer A may become infected in turn.



Trust table at Peer A

Peer	Trust value	Infectious value
B	0.7	2
C	0.6	1
D	0.5	0

Fig. 1. Example of the proposed scheme using a double-layer trust management

A. Trust Model

In the trust model, there are N peers, where each peer has a trust table with $2 \times (N-1)$ entries. The trust value and the infectious value in the trust table are used to select the downloading source. The trust model has the following major components.

- Trust table. The trust table in peer i is denoted as $T(i)$. The trust value of peer i on peer j, is denoted as $T_v(i,j)$, where $T_v(i,j) \in [0,1]$. For example, $T_v(i,j) = 0$ means that peer i has no trust for peer j and any filed downloaded from j would expected to be infected with probability 1.0. On the other hand $T_v(i,j) = 1$ means that peer i trusts peer j and any file downloaded from j is expected to be innocuous with probability 1.0. Therefore, in the selection of the downloading source, peer j has top priority to become the downloading source. Peer i updates his trust table after downloading a file from peer j by re-evaluating its trust and infectious values about peer j according to the experienced actions in the interaction with peer j and these are calculated as the fraction of downloads of clean files among all downloads from peer j. The design here complies with a well-known trust decay rule: a person tends to have higher trust on those referred by his/her immediate friends than on those referred by friends of friends. The difference of the friendship type defines the term *social distance*. Therefore, in general, the trust value decreases proportionally

to the *social distance* between peers. The second value in $T(i)$ is the infectious value I_v that represents the possible internal infection degree of peer j. The higher the value of I_v, the higher the possibility that a file, whether the host peer is known having viruses and with or without record of providing infected files contains malware. If there are several trustees who raise above the threshold for an acceptable trust value, the peer with the smallest I_v is selected as the downloading source. Peer i updates $T(i)$ if it receives an alert from its trustee, peer j.

- Antivirus software. Herein, it is considered that a peer has virus-detection software available. A successful virus detection indicates that a peer has downloaded an infected file, and the antivirus software can identify the file. Therefore, peer i detects a virus with probability $P_d(i)$.
- Internal infection. If a healthy peer (i.e., a host whose files are virus free) downloads a file containing viruses, other existing files in this peer can possibly get infected with probability P_I. An infected download is defined as a download of a file containing a virus.
- Propagation delay. The propagation delay is the time to download a file or the time used for dissemination of trust values. This delay is proportional to the distance between the source and the destination, given in number of hops (however, the delay can be use time units for actual implementations). The propagation delay between peer i and peer j is denoted as $d(i, j)$.
- Alert Buffer. In each peer, a memory block is utilized to store the received alerts from other peers. As soon as the alert is processed, the alert is cleaned from the peer to reduce the storage.

B. Management Scheme

The trust management scheme works as follows:

- Step1. When peer i searches for a file, it sends a request to its possible trustees whose trust value is larger than an admissible threshold Th_T. Peer i chooses one among all peers who respond positively (i.e., hosting a copy of the requested file) with the lowest infectious value as the download source. The peer with the smallest infectious value is selected to reduce the possibilities of downloading a unknown infected file.
  If no matched files are found, peer i sends a help query to all its trustees. When a peer receives a help query, it recursively searches for the requested file with the help of their own trustees. As an example, suppose that peer j is trustee of peer i and peer k is the trustee of peer j. The trust value of peer i on peer k is calculated as $T_v(i, j) \times T(j, k)$. In addition, the infectious value of peer i on peer j is expressed as $I_v(i, j) + I(j, k)$. Among all the trustee candidates whose T_v is above Th_T, peer j selects the one with the lowest infectious value.
- Step 2. Peer i stores the downloading history in its memory block. The trust value for peer j is calculated as the total number of clean downloads from peer j over the total number of downloads. After each downloading, peer i re-evaluates its trust value on peer j according to the download success rate.
  When an infection occurs, peer i computes a new trust value $T_v(i, j)'$ in the trust table. The change of the trust value is expressed as $\psi(i, j) = T_v(i, j) - T_v(i, j)'$. At the same time, a new infectious value is calculated as $I_v(i, j) = I_v(i, j) + 1$ if $I_v(i, j) < I_v^{max}$, where I_v^{max} is the maximum infectious value.
- Step 3. If the trust value of peer i on peer j has a significant drop, such that $\psi(i, j)$ is larger than the *warning threshold*, alerts are sent out to peer i's trusters. By this way, the possible proliferation of malware can be suppressed rapidly. Peer i sends the alert message in the following format: $\{ID, v_j, f, \Delta, d\}$, where ID is used for uniquely marking the alert including the source id, v_j is the subject host id, f is the name of the malicious file, Δ indicates the magnitude of shrink of the trust value, and d is the maximum number of hops that the alert is allowed to propagate.
  In this way, peer i broadcasts the alert warning to all its trusters. The propagation delay of the alerts is proportional to the distance between the peer i and all the destinations. When a peer receives an alert from peer i, it updates the relevant trust value and infectious value accordingly and may propagate this alert to its trusters. Generally, when peer k receives the alert from peer i, peer k makes decisions after checking the alert's content. If $v_j \notin \Theta(k)$, where $\Theta(k)$ is the set of k's trustees, the alert is dropped. Otherwise, a new trust value of peer v_j at peer k is calculated $T_v(k, j)' = T_v(k, j) - \Psi \times T(i, j)$ and the infectious value is updated as $I(k, j) = I(k, j) + 1$. If $\Psi \times T(i, j) <$ *warning threshold* or $d = 1$, the propagation of this alert is decreased. Otherwise, peer k sends the alerts to its trusters.

III. Performance Analysis

We simulated a P2P network using a mesh topology, with 100 nodes selected randomly as active peers in the mesh. An active peer is a host that forwards, stores, or requests files to or from the other peers. The network has 150 existing files with several copies for each file. Files (and copies) are distributed randomly with a uniform distribution among peers. From these file, we consider that 60% are popular files (i.e., requested with high frequency). Among all files, 10 randomly selected files are flagged as malware (i.e., virus). After a host downloads a malicious file, there is a probability of detecting it, denoted as P_d. Here, we consider that the minimum time for an event (e.g., a download or a transmission of an alert from one node to another in the network) is a fixed amount of time or time slot. We evaluate the total number of infected peers after each time slot.

Figure 2 shows the performance of the DTM scheme, where only a trust value per node is used. In this scheme, the trust value of a node is only evaluated by considering the download records of a truster about its trustees. This figure considers no delays when a file is downloaded nor when peers' trust values are broadcast to trusters, and $P_I = 0$. The figure shows two curves, one with $P_d = 0.5$ and the other with $P_d = 0.25$. Because the number of infected peers changes differently time

slot by time slot, the curve for $P_d = 0.25$ converges to 70 infected nodes after 20 timeslots, while the curve for $P_d = 0.5$ converges to 50 nodes after 20 time slots. This shows that the management scheme cannot bound the malware proliferation efficiently even without considering local infection at a peer.

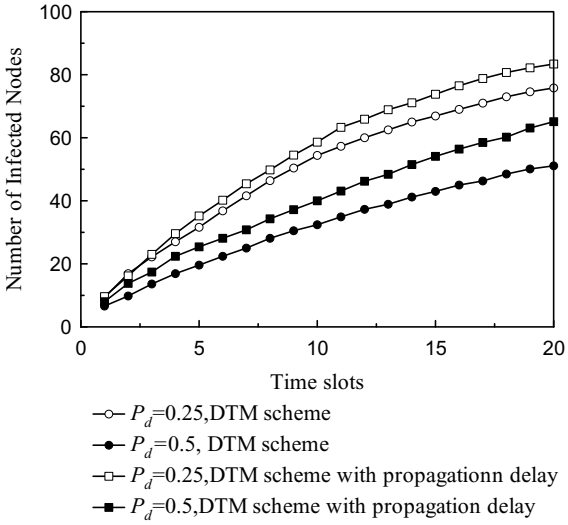


Fig. 2. Proliferation of malware using T_v and $P_d = \{0.25, 0.5\}$, with no local infection and alert delay.

This figure also shows the performance of the DTM scheme with $P_I = 0.0$ in a network. When peers use antivirus software with $P_d = 0.5$, the maximum number of infected peers approaches 45 after a period of time (or until the number of downloads reaches 800), and with $P_d = 0.25$, the maximum number of infected peers approaches 70 after 750 downloads. Delay on disseminating the alert messages in peers would give time for performing more infected downloads, producing heavy proliferation of malware.

Figure 3 shows the proliferation of the DTM scheme, as in the two cases above but, however, with local infection ($P_I = 0, 0.25, 0.5$). This case also considers no propagation delay for the alert system and $P_d = 0.5$. This figure shows that the local infection increases the effectiveness of malware proliferation and even with peers having $P_d = 0.5$, all peers in the network would be infected after 1200 downloads.

Figure 4 shows the degree of proliferation of malware using the DTM scheme and our proposed DDT scheme where I_v is used, under $P_d = 0.5$ with no propagation delay for distributing the alert messages. This figure shows the spreading of the malware in number of infected hosts per time slots. In this figure, the performance of the DTM scheme decreases as the P_I increases, i.e., the chances of malware infecting other files in the same malware host, making the malware resistant to that management scheme. On the other hand, with the proposed DDT scheme, the impact of the infection probability is also noticeable but this impact is significantly lower, making the proposed scheme more effective.

These results are also shown in terms of the number of downloads. Figure 5 shows the proliferation of malware using the DTM scheme and the DDT scheme under $P_d = 0.5$ with propagation delays for the alert messages. The proposed

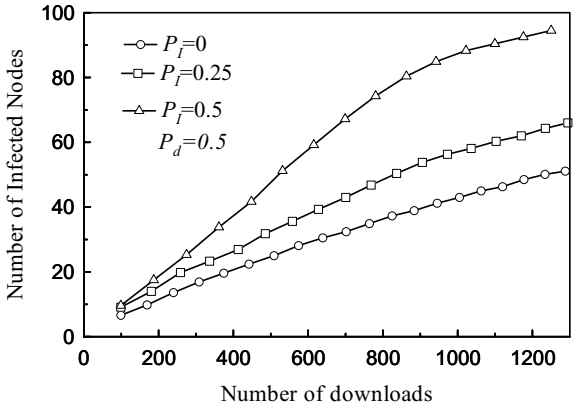


Fig. 3. Proliferation of malware using DTM scheme, with $P_d = 0.5$ and considering infection probability $P_I > 0$.

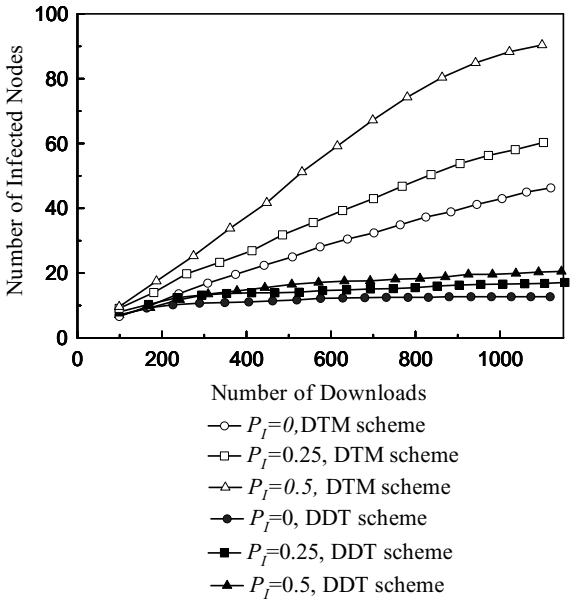


Fig. 4. Proliferation of malware using the proposed DDT scheme with $Pd = 0.5$ and different P_I values in time slots.

scheme not only inhibits the proliferation of malware but also bounds it. In the case of a high P_I value, or $P_I = 0.5$, the number of infected peers drops from 100 nodes as in the case of the DTM scheme to close to 30 peers in the DDT scheme.

Increasing the number of trust parameters in the management systems creates the risk of discouraging the download activity. The network's download activity (i.e., the number of performed downloads) was evaluated using the same conditions as above. Figure 6 shows the download activity of a network using the DDT scheme, in downloads per time slot. The results show that the download activity with different P_I values, which impacts I_v for each node, has no significant changes. This means that the proposed approach does not discourage network activity.

IV. Conclusions

Trust management is a promising strategy to bound the proliferation of malware on peer-to-peer networks that can

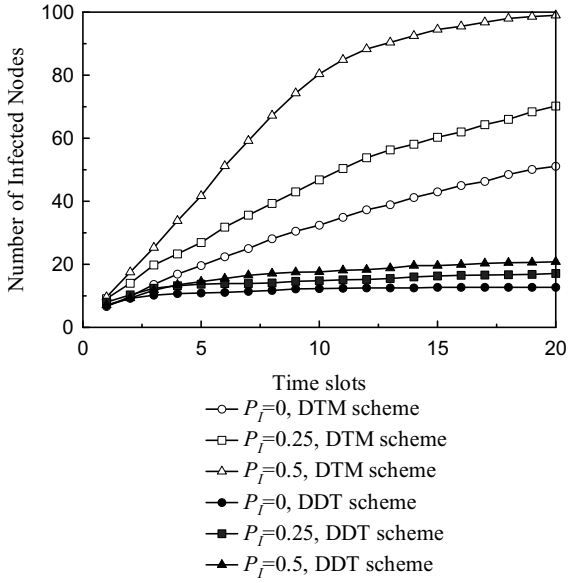


Fig. 5. Proliferation of malware using the proposed DDT with $Pd = 0.5$ and different P_I values.

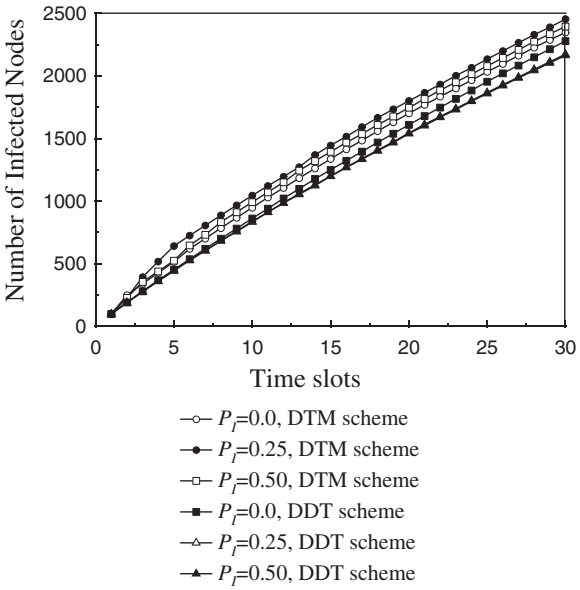


Fig. 6. Download activity of the network using the proposed DDT scheme.

work jointly with virus detection systems. In this paper, we showed that the use of a single trust value per peer has deficiencies in bounding the proliferation of malware. In most cases, it is highly probable that the majority of peers become infected. By using extra information, based on the infectious value, where the consideration of a peer having hosted an infected file, the proliferation of malware becomes bounded more effectively. By using computer simulation of a mesh peer-to-peer network we have shown the improvement of this proposed approach. Furthermore, considering that trust parameters to bound proliferation have the potential of discouraging download activity in P2P networks, we studied the impact of using our proposed DDT scheme. However, we showed that our approach has little impact on the download activity of the network.

REFERENCES

[1] X. Xu, Y. Wang, S. P. Panwar, and K. W. Ross, "A Peer-to-Peer Video-on-Demand System using Multiple Description Coding and Server Diversity," Proc. IEEE International Conference on Image Processing (ICIP), vol. 3, pp. 1759-1762, October 2004

[2] X. Hei, C. Liang, J. Liang, Y. Liu and K.W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," IEEE Trans. on Multimedia, 15 pages, December 2007.

[3] M. Macedonian, "Distributed File Sharing: Barbarians at the Gate?" IEEE Computer, Vol. 33, Issue 8, pp. 99-101, Aug. 2000.

[4] Y. Wang, X. Yun, Y. Li, "Analyzing the Characteristics of Gnutella Overlays," Proc. IEEE IV International Conference in Information Technology, 2007, pp. 1095-1100, 2-4 April, 2007.

[5] X. Zhang, S. D., and H-H. Chen, "Analysis of Virus and Antivirus Spreading Dynamics," Proc. IEEE Global Communications Conference (Globecom) 2005, Vol. 3, 5 pages, Nov. 28-Dec 2, 2005.

[6] P. Li, Z. Wang, X. Tan, "Characteristic Analysis of Virus Spreading in Ad Hoc Networks," Proc. IEEE Workshop in Computationall Intelligence and Security (WCIS) 2007, pp. 538-541, December 16-19, 2007.

[7] L-C. Chen and K.M. Carley, "The Impact of Countermeasure Propagation on the Prevalence of Computer Viruses," IEEE Trans. on System, Man, and Cibernetics, Vol. 34, issue 2, pp. 823-833, April 2004.

[8] E. Damiani, D. C. Vimercati, S. Paraboschi, P. Samarati, and F. Violante, "A Reputation-based Approach for choosing Reliable Resources in Peer-to-Peer Networks," Proc. of the 9th ACM conference on Computer and communications security (CCS), pp. 207-216, Washington, DC, November 2002.

[9] S. Marti and H. Garcia-Molina, "Limited Reputation Sharing in P2P Systems," Proc. of the 5th ACM Conference on Electronic commerce (EC), pp. 91-101, New York, NY, May 2004.

[10] J. Shin, T. Kim, Taehoon, and S. Tak, "A Reputation Management Scheme Improving the Trustworthiness of P2P Networks," Proc. IEEE International Conference on Convergence and Hybrid Information Technology (ICCHIT) 2008, pp. 92-97, August 28-30, 2008.

[11] X. Dong, W. Yu, and Y. Pan "A Dynamic Trust Management Scheme to Mitigate Malware Proliferation in P2P network," Proc. IEEE International Conference on Communications 2008, 5 pages, Beijing, China, May 2008.

[12] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim., "A Survey and Comparison of Peer-to-Peer Overlay Network Schemes," IEEE Comm. Survey and Tutorial, Vol. 7, Issue 2, 2^{nd} Quarter 2005, pp. 72-93, 2005.