Detecting Sybils in Peer-to-Peer File Replication Systems

K. Haribabu[1], Chittaranjan Hota[2], and Saravana[1]

[1] Computer Sc. & Information Systems Group, Birla Institute of Technology and Science Pilani, Rajasthan, India
[2] Computer Sc. & Information Syst. Group, Birla Institute of Technology and Science, Pilani Hyderabad Campus, Hyderabad, Andhra Pradesh, India
Khari@bits-pilani.ac.in, hota@bits-hyderabad.ac.in, saravana87@gmail.com

Abstract. The test of a peer-to-peer file sharing network is how efficiently the objects are discovered and retrieved. One of the most important factors that contribute towards this is optimal replication of the objects across the network. One of the security threats to replication model is Sybil attack. In this paper we propose an approach that aims at detecting sybil identities in peer-to-peer file sharing networks. The sybils can corrupt, hide or destroy the replicas in file sharing network. This approach makes use of the fact that sybil doesn't scale its storage to the factor of its identities. The approach safeguards the availability and accessibility of objects in a peer-to-peer network from sybil attack. Experimental evaluations have shown that our approach works very efficiently in detecting sybils. More than 50% of the sybils were detected in first few seconds of the simulation and loss or damage of objects is reduced to less than .0001%.

Keywords: Peer-to-Peer, Overlay Networks, Sybil Detection, Replication.

1 Introduction

P2P overlay networks are application-level logical networks built on top of the physical networks. These networks maintain separate addressing and routing mechanisms to enable efficient search and data exchange between peers. They don't require any special administrative or financial arrangement. They are self-organizing and adaptive, distributed and decentralized. P2P overlay networks are categorized as unstructured and structured. An unstructured P2P system is composed of peers joining the network with some loose rules, without any prior knowledge of the topology. Gnutella [1], and KaZaA [2] are examples of unstructured P2P overlay networks. In structured P2P overlay networks, network topology is tightly controlled and content is placed not at random peers but at specified locations that will make subsequent queries more efficient. Most of the structured P2P overlays are Distributed Hash Table (DHT) based. Content Addressable Network (CAN) [3], Chord [4], and Pastry [5] are some examples of structured P2P overlay networks.

When compared to P2P model, client/server model has smaller security risks due to the presence of centralized authority which can authenticate requesting nodes. P2P model, due to its decentralized model thus lacking centralized authority to authenticate nodes, faces security challenges such as masquerading, denial of service (DOS),

D. Weerasinghe (Ed.): ISDF 2009, LNICST 41, pp. 123–134, 2010.

and tampering [6]. By masquerading or acting as another node, a node can give misinformation to other nodes. This can lead to partitioning the network, DOS attack, and illegal actions on the network. A malicious node can repeatedly send requests for content from a particular node and thus preventing it from fulfilling legitimate requests. Another challenge is to protect content that is replicated on different nodes from alterations. Powerful attacks such as Sybil [7], Byzantine agreement disruption [8], and DOS [8] attacks make the large part of the network fail. Routing in P2P networks involves passing messages to intermediate nodes which can be malicious. Secure routing deals with malicious nodes actively trying to disrupt communication. Since there is no central authority to verify the one to one mapping between identity and node, a node can masquerade as multiple identities. This way it can control large part of the network. This is called Sybil Attack [7]. Although this kind of attack is possible in P2P networks, ad-hoc networks and sensor networks, it can do major damage in P2P networks because of their large global size and lack of physical constraints as in ad hoc and sensor networks [9]. It is not very difficult to set up this attack because it requires one node and many different identities or abstractions.

This attack can disrupt the network message routing for look up process and overlay maintenance by controlling large portion of the network. It is necessary to ensure that a message will be delivered to the intended node despite malicious node activity such as message corruption or misrouting [10]. A Sybil has the power to actively monitor and control the ongoing unicast communications or collaborative agreement tasks. The specific attacks that can be mounted are impersonating different routes and controlling them, dividing or segmenting a part of the overlay by positioning itself conveniently, mounting a denial of service attack on a particular node, or disrupting a Byzantine agreement.

Sybil attack is the attack against identity where an entity in a network can masquerade itself as multiple simultaneous identities in the network. This problem is pervasive in all distributed systems. In real world election scenario, people can rig the elections or in other words they represent themselves forcefully on behalf of many people. Using this attack, companies increase the rank of the web pages in Google search results[11] and some people associate certain search terms with popular personalities out of fun[12]. The peer-to-peer systems are used for many purposes; computational [13-14], messaging [15], file sharing [1]; most popularly used in file sharing networks. Sybil attack has its affect on file sharing systems especially in replication of copies of files. By knowing the mechanism of replication which is used in a particular P2P network, a malicious user (Sybil attacker) can create fake identities in the network so that the file replication of a particular file happens entirely or partially happens on the Sybil identities created by this particular user. Once the replica is in the hands of Sybil identity, it can corrupt, hide or destroy the copy what to speak of when all copies are replicated on Sybil identities only. Sybil attack goes against maintaining quality and accessibility of content, and robustness of the network. In this paper, we address this problem by developing a protocol that is capable of detecting Sybil identities.

The simulation results show that this approach can detect Sybil identities to the degree that loss of file replicas are reduced to less than .0001%. However, this approach is less efficient when the number of replicas being maintained is less. Having very few replicas of the objects, which is of course an unlike scenario in today's peer-to-peer file sharing systems.

2 Related Work

Douceur [7] describes puzzle methods that exploit communication, storage or computational resource constraints. He proves that computational puzzle methods are not viable. In these puzzles, the verifier sends a large random value to every other identity it wants to verify. These identities must then compute the solution within a constrained amount of time. If an entity has more than one identity it will fail to compute the solution within the time. The paper says that this can be circumvented by taking help of other powerful nodes. Thus he advocates the existence of a central authority to prevent Sybil attacks. Castro, et al. [10] argue that in a P2P overlay network, if a central authority distributes uniform node identifiers (IDs) then it is difficult for attackers to have any control over this process. They allow multiple node IDs per IP address. Dinger and Hartenstein [9] proposed an identity registration procedure called self-registration that is a natural extension of P2P mechanism to safeguard against Sybil attacks. Their approach clearly distinguishes network nodes from participants. The results of their self-registration process show that it is able to regulate number of nodes per participant. It has open-ended questions like within what duration the network becomes free from dominance. Danezis, et al. [16] present a modified DHT routing using a bootstrap tree for Chord to resist the impact of Sybil attacks. In the bootstrap tree, two nodes share an edge if one introduced the other into the DHT. These relationships are established outside the tree off-line. With this logic Sybil nodes will attach to the tree at limited nodes. Also, a trust metric is used to minimize the probability of a malicious node being on the routing path of the Chord. Fiat, et al. [17] proposed S-Chord which is a variant of Chord where a node uses a set of nodes called Swarm which randomly selects an ID using which the node positions itself at a critical position in the Chord ring. In Sybilguard [19], the authors have proposed a distributed algorithm for limiting entry of Sybil identities into a social network. They have used the principle that in a social network, the trusted edges between honest group and a Sybil group will be very few. They have designed a protocol in which the verification of new entry into the network is done by intersection of random routes.

Our approach is based on challenging resources of the Sybil identities. The approaches [20-21] also fall into the same category. Unlike the other challenge-resource approaches, this approach is more reliable because the storage is persistent. Here it is not difficult to simultaneously test the storage capacity of most identities because it can be done over a period of time.

3 Sybil Detection

In this section we discuss design of our approach.

3.1 Scope

File replication in P2P has many advantages such as reducing traffic congestion; increasing object availability and aiding in fault tolerance. Single node failures, like crashes of nodes, can be tolerated as faults within the system as a whole facilitated with the help of the redundancy introduced by replicas. If a host of a replica fails,

requestors may access another host with a replica. Data replicated at more than one site facilitate to minimize the number of hops before the data are found. But, large scale Peer to Peer systems face security threats from faulty or hostile remote computing elements [7]. Peer-to-Peer (P2P) based file sharing applications have become highly popular in today's Internet due to the spread of platforms such as Napster, Gnutella, KaZaa, eDonkey, BitTorrent, and others. The Sybil attack in which a single user can pose as multiple identities is a serious threat to P2P File Sharing Systems because the malicious entity can sabotage the P2P file sharing system in whatever way he likes. The various ways in which a Sybil Node can attack or disrupt the functioning of the file sharing networks are given below:

Content Pollution: The Sybil identities can behave in various ways such as replacing all or part of the content with white noise, cutting the duration, shuffling blocks of bytes within the digital recording, inserting warnings of the illegality of file sharing in the recording, and inserting advertisements; the main aim being to render the file unusable and thereby reducing its popularity [18]. Now, this polluted content can be replicated on a large number of honest or Sybil nodes in the P2P Network. A normal user who is oblivious to all these, downloads these content and thus the polluted content spreads throughout the file sharing network eventually exceeding the number of original copies. As the users download more and more polluted copies, it might lead to frustration among users and subsequently leading them to abandon the file sharing itself. A situation can happen is that when a recording company is on the verge of releasing a song that will likely be popular; the rival record company might pay a pollution company to spread bogus copies of the song through one or more P2P network thereby reducing the popularity of the file.

Content Deletion: Consider the other side of file replication on multiple nodes; it introduces a new set of challenges. The Sybil identities on which the files are replicated might delete the files that were replicated and be detrimental towards to the file sharing system. For example, a person X may store his audio or video at a remote node Y located elsewhere so that persons near that can download the files from that particular node. If that node was a Sybil node, the file might be deleted and the basic purpose of replication in file sharing systems is defeated.

Content Concealment: The Sybil node can possess the file and not send it to the requesting node. By this way, again the motives of the P2P file sharing network reduces because of congestion etc. In this case, the Sybil Identity might still possess the data (so that if the owner node verifies, it would be able to resend the data and confirm it) and thus conceal it from other requesting nodes.

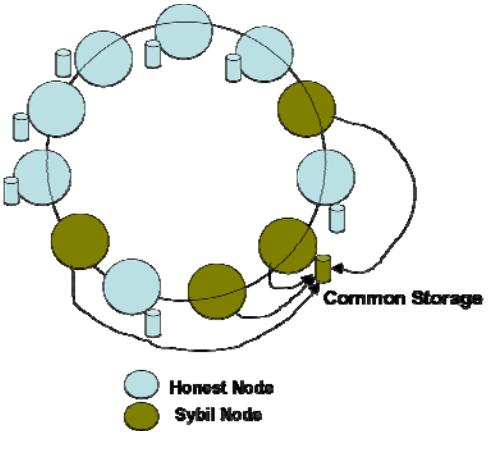

Fig. 1. Sybil identities having common storage in Chord network

Therefore, the effects of Sybil Identities can be devastating to a P2P file sharing network. So, it is advisable that files are not replicated wholly on Sybil Identities. Towards this SybilGuard [19] mentions that maintain g+1 replicas in case there are g Sybil groups is wasteful and instead propose a mechanism which bounds the number of nodes in each Sybil Group. Whereas, in our work we don't present any restrictions on the minimum number of replicas that need to be maintained as our theory works even if all the files are replicated wholly or partially on the Sybil Identities.

In this paper, we attempt to solve this problem, by a two phase method. 1. Detecting Sybil identities in the process of replication 2. Adapting replication process so that the replicas will not go to already detected Sybil identities

3.2 Algorithm Overview

As we have seen that Sybil attack creates multiple simultaneous virtual identities in the network. Virtual identities mean that although these identities created by the user appear to be normal, they don't have their own computational, memory and storage resources. But for the nodes in the network, these identities appear to be no different than normal nodes. Literature describes various ways to detect Sybils. One of the ways is to detect whether the node is a virtual or just a normal node.

The network is a structured network where the nodes are placed in pre-determined positions according to the node id. The data is placed in the node whose id is the closest to the key of the object. In most structured networks, the objects are replicated in r number of successors, r being dependent on individual system. The node where the object is originally stored is called 'owner' of the object. The owner replicates the copies of the object. The owner of the file has details about to which all nodes the file has been replicated. For all the Sybil identities created by one particular malicious user the storage would be done in one particular place i.e., the malicious user's storage system. So, the data that has been replicated in Sybil identities created by that malicious user will be stored in the storage area of the malicious user. The malicious user has a limited storage capacity. As the number of Sybil Identities on which file replication takes place increases, the storage capacity needs to be increased and the cost involved in increasing storage compared to the benefit in doing that proves ineffective for the malicious user as the data which is replicated is not a highly confidential data in normal P2P networks.

The owner of the object replicates a file on a set of nodes. The owner needs to keep track of the node identities where it has replicated the files. It is assumed that for a normal node, there will not be a situation where it doesn't have space to store the file. This is because the replication placement is done by the consistent hashing [22]. This will ensure equal distribution of load.

Every owner of the object verifies the existence of the files at regular intervals. The owner sends a message to each node where the replica is placed asking for a randomly chosen byte range within the file. The receiver is supposed to send the reply with those few bytes extracted from the object. When the verifier receives a reply, it verifies the reply. If either the reply is not received or the reply is not correct then, the owner notes the node identifier of the node. If the same situation occurs for a more than threshold

number of times, the owner detects the node to be a Sybil identity. Then it will not replicate the objects anymore on this node.

The verification message consists of {fileId, fromByteOffset, toByteOffset}. The verification reply message consists of {fileId, fromByteOffset, toByteOffset, bytes}. After sending the verification request, the owner waits for the reply. If the reply is not received in an expected amount of time, the owner takes it to be a no reply. The replication and verification procedures are outlined in Fig2 and Fig3.

```
Replicate(File:f)
{
   Successors: S
   SybilDetected: D
   Int: noOfReplica=0
   ReplicaList: L
   for each successor s  ∈ S
     if (s  ∈ D) = false then
        putIntoNetwork(s, f)
       add(L, s, f)
     noOfReplica = noOfReplica+1
     if noOfReplica>REPLICA_LIMIT then
   exit for
}
```

Fig. 2. Algorithm for replicating a file f

```
VerifyReplications(ReplicaList:L)
{
   SybilDetected: S
   ReplicaList: L
   VerificationMessage: v
   VerificationReplyMessage: vr

   for each replica r  ∈ L
      v= makeVerificationMessage(r)
      sendVerificationMessage(r.s, v)
      vr = waitForReply()
  if verifyReply(r, v, vr)==false or
vr==null then
       L.noReplyCount =    L.noReplyCount
+ 1;
  If L.noReplyCount > THRESHOLD then
     S.add(r.s)
}
```

Fig. 3. Algorithm for verification of a copy of file f

```
VerificationMessage
makeVerificationMessage(Replica: r)
{
     Int: fromByte
     Int: toByte
    VerificationMessage: v

   fromByte=getRandomNo() Mod sizeof(r.f)
   toByte= getRandomNo() Mod sizeof(r.f)
  if fromByte>toByte then
      swap(fromByte,toByte)

   v.f = r.f
   v.fromByte = fromByte;
   v.toByte = toByte;
    return v
}
```

Fig. 4. Procedure for making a verification message

```
verifyReply(Replica: r,
VerificationMessage: v,
VerificationReplyMessage: vr)
{
   If byterange(r.f, v.fromByte, v.toByte)
== vr.data  then
       return true
  else
      return false;
}
```

Fig. 5. Procedure for verifying a reply sent from a node

4 Simulation Results

Simulation was carried out on a 1000-node Chord network. We used PlanetSim [23] overlay network simulator. Necessary changes were made in the Node classes to represent the current purpose of simulation. New procedures were written for replication and verification. The simulator was a step based simulator. Every step, the messages are transferred from current node to next node. The simulation was carried out for 45000 steps. The files are replicated in the system throughout the simulation using a Poisson process with average as 4. The threshold value for terming a node as

Sybil is 4. The waiting time for a verification reply is set to 10 seconds. The topology of the Chord network is shown in Fig 6.

In the beginning of the simulation, all the honest and Sybil nodes are created. The honest nodes are 1000 in number. The Sybil nodes are varied from 50 to 850 i.e. 4.7% to 46%. We see in graph Fig 7 that all the graphs follow the same pattern. Initially all the curves are steeply falling, indicating that there is high probability that the objects are distributed to Sybil identities but since there is no storage space, they could not hold all the replicas. As the number of Sybil identities reduce in the network, the probability that a object is replicated in a Sybil node also reduces. That is why the steepness of the curves reduces. Also we can observe that as the Sybil identities percent in the network is increased, the time taken to detect Sybil identities also increases. In Fig 8, we can observe that, as the percent of Sybil identities increase in the system, the total number of Sybil identities detected in 45000 steps is reduced. In Fig 9, we see that reducing the number of Sybil identities has direct effect on file losses incurred in the network. We can see from the Sybil CDF that when it has reached a slow progress state, accordingly the file losses also have reduced. Normally the file losses are due to the Sybil identities, since they don't have the storage space to store replicas of all the Sybil identities. When they are detected, the files are replicated on a different set of nodes probably honest nodes. That way the file replicas are safer. In Fig 10 we can see how the Sybil detection procedure is dependent on the number of files being replicated in the network. The whole algorithm is dependent on the replicas of files. More the number of files replicated, more will be the detection of the Sybil identities. In Fig 11, it can be observed that, the waiting time for a verification reply from a node has no drastic influence of the detection of Sybil identities. This is because several nodes replicate their objects on a Sybil node. The increase in waiting time doesn't delay the detection because there are several other nodes which are verifying meanwhile.

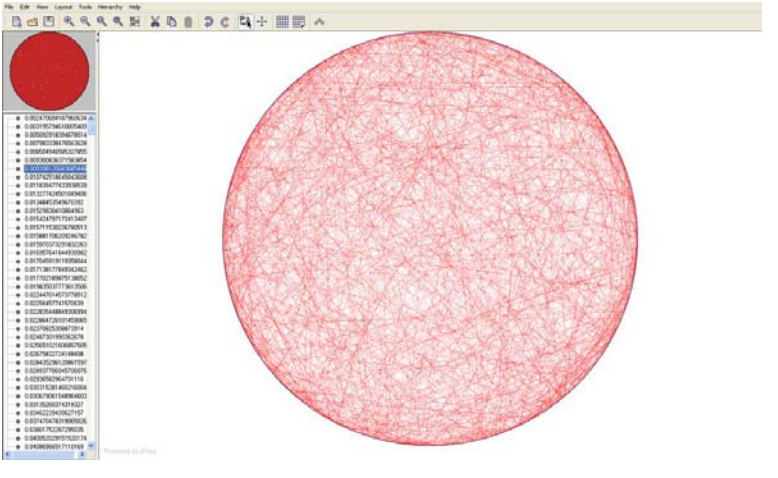

Fig. 6. Chord network topology for 1000 node network

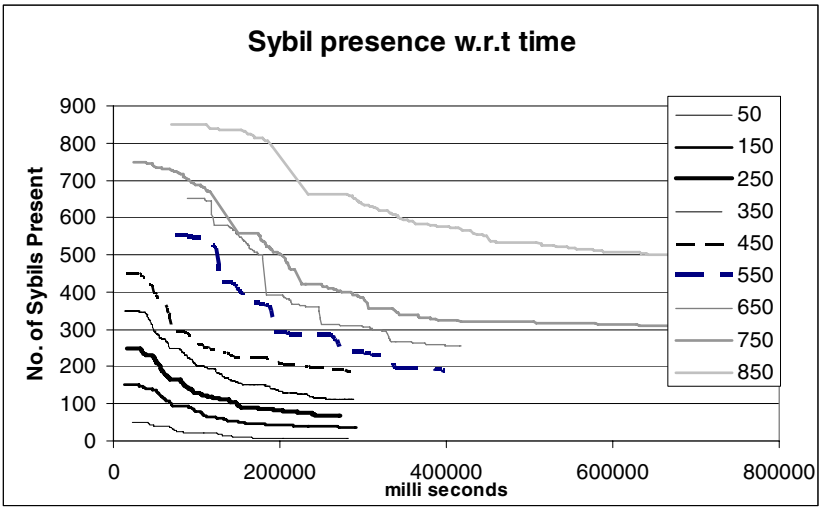

Fig. 7. Detection pattern of Sybils for different % of Sybils in the network

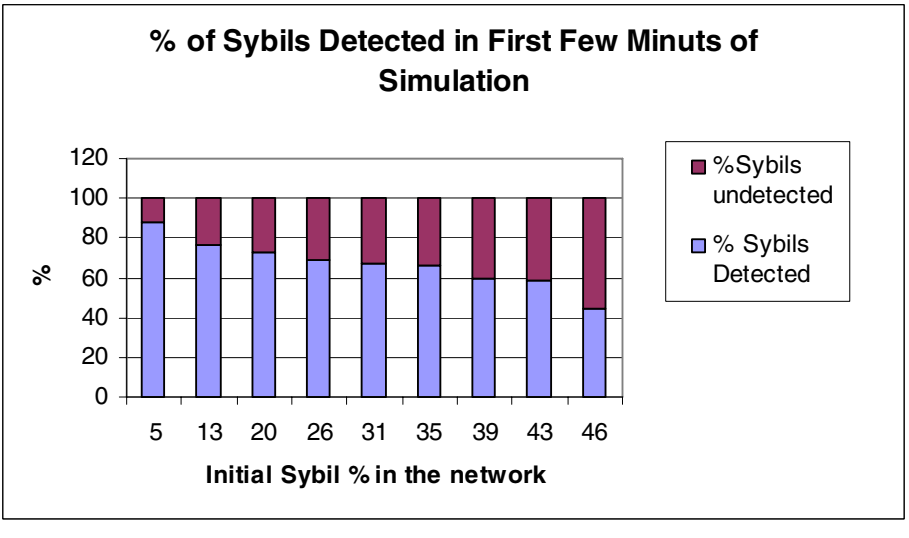

Fig. 8. Effect of % of Sybils on detection algorithm

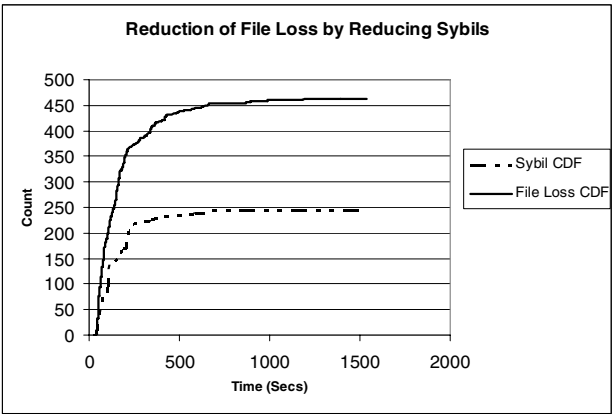

Fig. 9. Effect of Sybil detection on replica losses

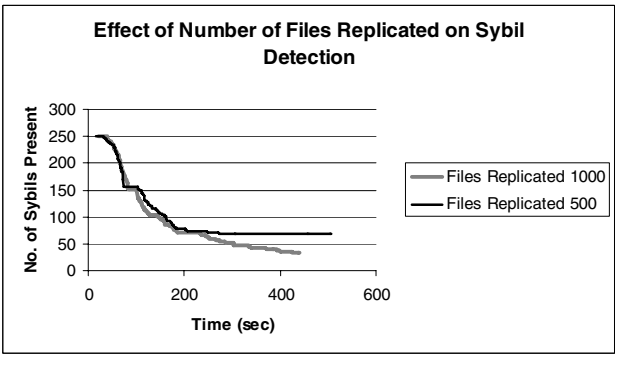

Fig. 10. Sybil detection depends on number of object copies replicated in the network

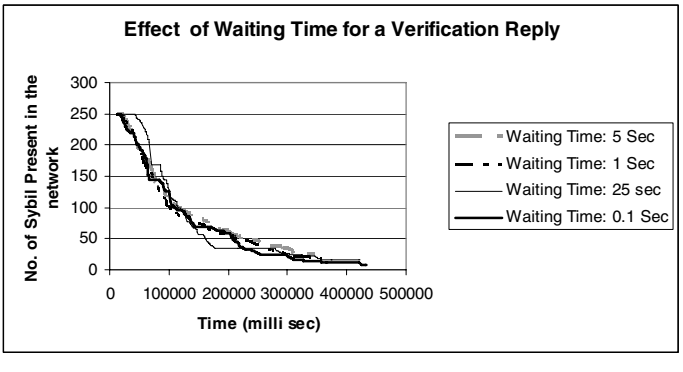

Fig. 11. The effect of verification waiting time on the Sybil detection is almost nil

5 Conclusion

This paper presented a novel decentralized protocol for limiting the corruptive influence of Sybil attacks on replication system in peer-to-peer networks by detecting Sybil identities and there by avoiding storing replicas on them. This approach relies on the principle that Sybil doesn't scale its storage capacity to the factor of its identities. Also unlike the other challenge-response approaches, this approach is more reliable because the storage is persistent. Here it is not difficult to simultaneously test the storage capacity of most identities because it can be done over a period of time. Experimental evaluations on this approach have shown that Sybil identities were detected to the extent of 90% of initial Sybil identities. Also the effect of parameters like initial percent of Sybil identities, total number of objects replicated in the network, waiting time for a verification reply is analyzed. Still the approach may suffer if the Sybil identities chose to store the replicas on another node. Our future work will focus on a fool proof 100% Sybil detection protocol with simulations on a larger network.

References

1. Gnutella Protocol Specification Version 0.4, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf
2. Kazaa, http://www.kazaa.com
3. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A Scalable Content Addressable Network. In: Proceedings of the 2001 ACM Annual Conference of the Special Interest Group on Data Communication (SIGCOMM), pp. 161–172. ACM Press, New York (2001)
4. Stoica, I., Morris, R., Liben-Nowell, D., Karger, D., Kaashoek, M.F., Dabek, F., Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. IEEE/ACM Transactions on Networking 11, 17–32 (2003)
5. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–350. Springer, Heidelberg (2001)
6. Haiying, S., Brodie, A.S., Xu, C., Shi, W.: Scalable and Secure P2P Overlay Networks. In: Wu, J. (ed.) Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks. CRC Press, London (2005)
7. Douceur, J.R.: The Sybil attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)
8. Wallach, D.S.: A Survey of Peer-to-Peer Security Issues. In: Okada, M., Pierce, B.C., Scedrov, A., Tokuda, H., Yonezawa, A. (eds.) ISSS 2002. LNCS, vol. 2609, pp. 253–258. Springer, Heidelberg (2003)
9. Dinger, J., Hartenstein, H.: Defending the Sybil Attack in P2P Networks: Taxonomy, Challenges, and a Proposal for Self-Registration. In: Proceedings of the First International Conference on Availability, Reliability and Security (ARES 2006), pp. 756–763. IEEE Computer Society, Los Alamitos (2006)
10. Castro, M., Druschel, P., Ganesh, A., Rowstron, A., Wallach, D.S.: Secure routing for structured peer-to-peer overlay networks. In: Proceedings of the 5th USENIX Symposium on Operating Systems Design and Implementation, pp. 299–314. ACM Press, New York (2003)

11. Bianchini, M., Gori, M., Scarselli, F.: Inside page rank. ACM Transactions on Internet Technology 5(1), 92–128 (2005)
12. Viglucci, A., Tanfani, J., Getter, L.: Herald special report: Dubious tactics tilted mayoral votes. Miami Herald, February 8 (1998)
13. Anderson, D.: SETI@home in Peer-to-Peer: Harnessing the Benefit of a Disruptive Technology. O'Reilly & Associates, CA (2001)
14. Larson, S.M., Snow, C.D., Shirts, M., Pande, V.S.: FOLDING@home and GENOME@home: Using distributed computing to tackle previously intractable problems in computational biology. Computational Genomics (2002)
15. Miller, J.: Jabber: Conversational technologies in Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology. O'Reilly & Associates, CA (2001)
16. Danezis, G., Lesniewski-Laas, C., Kaashoek, M.F., Anderson, R.: Sybil-resistant DHT routing. In: di Vimercati, S.d.C., Syverson, P.F., Gollmann, D. (eds.) ESORICS 2005. LNCS, vol. 3679, pp. 305–318. Springer, Heidelberg (2005)
17. Fiat, A., Saia, J., Young, M.: Making Chord Robust to Byzantine Attacks. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 803–814. Springer, Heidelberg (2005)
18. Liang, J., Kumar, R., Xi, Y., Ross, K.: Pollution in P2P file sharing systems. In: Proceedings of. IEEE INFOCOM 2005, vol. 2, pp. 1174–1185. IEEE Computer Society, Washington (2005)
19. Yu, H., Kaminsky, M., Gibbons, P.B., Flaxman, A.: SybilGuard: Defending against sybil attacks via social networks. In: Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 267–278. ACM Press, New York (2006)
20. Borisov, N.: Computational Puzzles as Sybil Defenses. In: Proceedings of the Sixth IEEE International Conference on Peer-to-Peer Computing, pp. 171–176. IEEE Computer Society, Washington (2006)
21. Aspnes, J., Jackson, C., Krishnamurthy, A.: Exposing computationally challenged Byzantine impostors. Technical Report, Yale University Department of Computer Science (July 2005)
22. Karger, D., Lehman, E., Leighton, F., Levine, M., Lewin, D., Panigrahy, R.: Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing, pp. 654–663. ACM Press, New York (1997)
23. García, P., Pairot, C., Mondejar, R., Pujol, J., Tejedor, H., Rallo, R.: PlanetSim: A New Overlay Network Simulation Framework. In: Proceedings of 19th IEEE International Conference on Automated Software Engineering, pp. 123–136. IEEE Computer Society, Los Alamitos (2004)