Enhancing Secure File Transfer by Analyzing Repeated Server Based Strategy using Gargantuan Peers (G-peers)

Kaushik Sekaran
Assistant Professor
School of Computing Science
& Engineering
VIT University, Vellore, India.

ABSTRACT
Peer-to-peer networks are useful for the file transferring within the group of peers in the network. But usually files are transferred in slow speed, if the files are larger in size. Also the peers don't have much utility within them. In order to enhance the utilities of the individual peers, each peer has to configure with greater CPU power, bandwidth, and storage space. But practically it is an impossible task, as the cost goes high by configuring peers with higher configuration. So my approach is to monitor the network carefully and to find out some random peers and to configure, say in a hundred peer attached network, one peer has to be configured with higher configuration. That will enhance the transfer speed among all peers. I introduce a new term called "G-peers", gargantuan peers, which is having all the utility with it. These G-peers are useful in securing and transferring files at a higher speed and enhance the overall performance of the network. By adding repeated G-peers in the network, it is notable that it will increase the speed of file transfer and also the files are much secured as we are transferring using G-peers authentication. Therefore, a high success rate can be achieved within the Peer-to-peer network by these G-peers.

General Terms
Peer-to-peer networks, file transfer and higher configuration.

Keywords
Gargantuan peers (G-peers).

1. INTRODUCTION
Peer-to-peer networks are used in the Internet to share resources (i.e. CPU power, bandwidth and storage space) between numerous peers. Resources are consequently distributed all over the Peer-to-peer network. The benefits of this approach are the efficient use of existing resources, scalability and consistency of the files shared between peers. The original intent of my work is to secure the files through proper authentication mechanism in the client (pool of peers) and also retrieving the files from the servers (G-peers), using the G-peers search Algorithm. In order to avoid the flooding and buffer overflow of the queries from the different peers, here I'm changing the G-peers in a random interval according to its DHT's index rank using G-peer Ranking technique.

2. BACKGROUND STUDY

2.1 Literature review
In conventional centralized networks there will low number of servers which provides core value to application, whereas in the peer to peer networks uses dissimilar connectivity among nodes in the network and collective bandwidth of participants in the network. Mostly peer-peer networks are used for establishing connection among the nodes via ad hoc connectivity. One of the true peer - peer networks are Gnutella [2, 3, 4] or Freenet [18] which is greatly assisted by directory servers that intimate the network address of the peers to other peer in the network.

2.2 Classifications of Peer-to-peer Networks
The Peer-to-peer networks are classified into five types according to their degree of centralization [13].

- Centralized Peer-to-peer network
- Decentralized Peer-to-peer network
- Structured Peer-to-peer network
- Unstructured Peer-to-peer network
- Hybrid Peer-to-peer network

2.3 File Sharing in Peer-to-peer Networks
File sharing [1, 20] is based on peer-to-peer model, where all the files are stored and served by users system. Most people who indulge in file sharing using Internet provide upload and download file. File trading [14] and Peer-to-peer file sharing are different, one mentioned later does not require uploading, but still some networks provide option for uploading such as credits or enforcing the sharing of files currently being downloaded. Apart from the traditional file sharing there are services [5] that send streams instead of files over a Peer-to-peer network. Hence anyone will be able to hear radio and watch television without the need of server. It is important that instead of a tree structure network [9, 11], a brimful technology from BitTorrent [19] is used.

2.4 Distributed Hash Table (DHT) [5, 6, 10]
While a peer queries for a certain file, then the file is searched for in an index structure maintained in every peer of the network. In order to maintain such an index structure we make use of a Distributed hash table. This Distributed hash table (DHT) serves as an index to satisfy the queries of other peers.

These are a class of decentralized distributed systems [21] that provide a lookup service [6] similar to a hash table: (name, value) pairs are stored in the DHT, and any participating node can efficiently retrieve the value associated with a given name. Responsibility for maintaining the mapping from names to values is distributed among the nodes, in such a way that a change in the set of participants causes a minimal amount of disruption. This allows DHTs to scale to extremely large numbers of nodes and to handle continual node arrivals, departures, and failures. DHTs form an infrastructure that can be used to build more complex services, such as distributed file systems, peer-to-peer file sharing and content distribution systems [8], cooperative web caching, multicast, anycast, domain name services, and instant messaging.

DHTs characteristically emphasize the following properties:

- Decentralization: The peers collectively form the system without any central coordination.
- Scalability: The system should function efficiently even with thousands or millions of nodes.
- Fault tolerance: The system should be reliable (in some sense) even with nodes continuously joining, leaving, and failing.

3. SELECTION OF G-PEER(S)

The below pseudocode has been applied for selecting G-peer in a group of peers. Initially, the peer with higher configuration would be taken as G-peer. So the demand once comes to G-peer from the queried peer is automatically transferred to DHT (Distributed Hash Table), for searching [7, 15, 17] the similar file in the network. If the file is not found in the first search by G-peer(1), the query is re-routed to the succeeding G-peer.

3.1 Pseudocode

```
While Pi is an G-peer
      If Qi is in DHTi then
         Send Ri as ACK to Qi
           Else
                  Pj communicates Qi to network
                        If Qi is in DHTk then
                         Qi resolved to Pj
                           Increment Rank of Pk
                            Make K as G-peer-'n'
                               Decrement Rank of Pi
                           Make I as Normal peer
                        End if
       End if
End while
```

4. SYSTEM DESIGN

4.1 Decision Flow chart

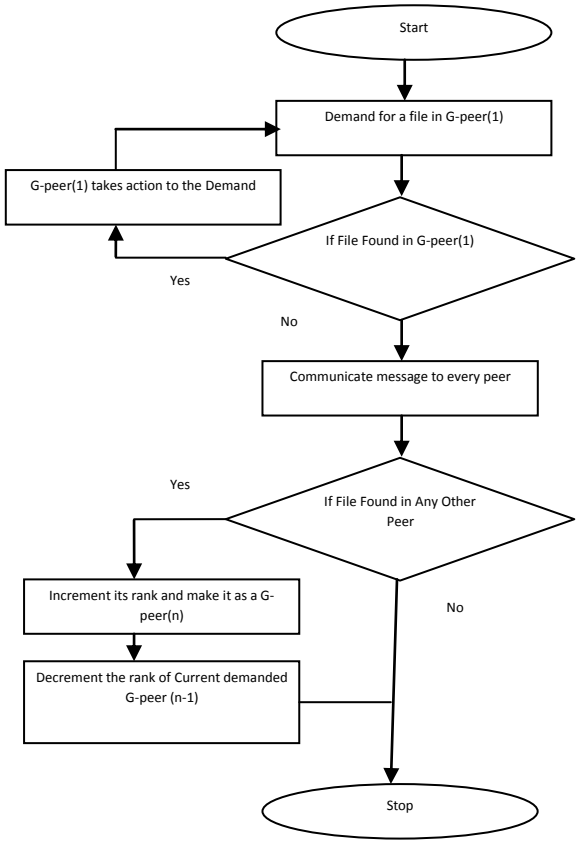

Fig 1: Decision Flow of G-peers search Algorithm

4.2 G-peers Search algorithm

In the Fig-1, the flow chart illustration clearly explains about the demand can be raised by any of the peer in the network and that file demand will be transferred to the G-peer index. If file found in G-peer-1, then it takes action .i.e., it will send that file to the Queried peer. If the file is not found by G-peer(1), then it will communicate the message to every peer in the network. Hence, here the message is being transmitted to every peer. If that particular file has been found in any other peer, then in the peers DHT, the rank is being incremented by 1. Also, the G-peer-1th rank is decremented by 1. Thus the queried file will be retrieved successfully using our G-peers search algorithm.

4.3 Design requirements

The design requirements for my approach are discussed below.

4.3.1 Security

In my work, security is important as each peer which enters the network should authenticate itself with the G-peer node.

4.3.2 Efficiency

The efficiency of the network is improved in this approach by using G-peers search algorithms.

4.3.3 Fault tolerance

Fault-tolerance or graceful degradation is the property that enables a system (often computer-based) to continue operating properly in the event of the failure of (or one or more faults within) some of its components. Usually Peer-to-peer networks is fully a fault tolerant networks i.e., even if a peer fails other peers in the network can able to share the files among them.

4.3.4 Scalability

Peer-to-peer networks are scalable networks [9, 16] as 'n' numbers of peers can be added to the network for file sharing utility.

5. IMPLEMENTATION

I have implemented my approach in a closed environment using Java language.

5.1 Sample coding

- For authentication with G-peer:

```
public void sendTgsInfo(String un,String serverName,String ts)
{
        /* Function used to add authentication tags to the peer queries. Authentication serves as an extended feature for securing the network */
}
```

- For Searching files in the G-peer:

```
public void listTextFilesInTree(File file)
 {
        /* used to search for a file queried from the DHT of an G-peer. Also supports Query forwarding [12] incase of a G-peer File Miss.*/
 }
public SendTgsInfo(String d,Vector v1,Vector v2)
{
/* Function used to establish Communication between peers */
}

SendInfo(Client cl1,String data1)
{
/* This Function is used to Send the file information to the respective peers*/
}
```

6. RESULT

6.1 Test screens

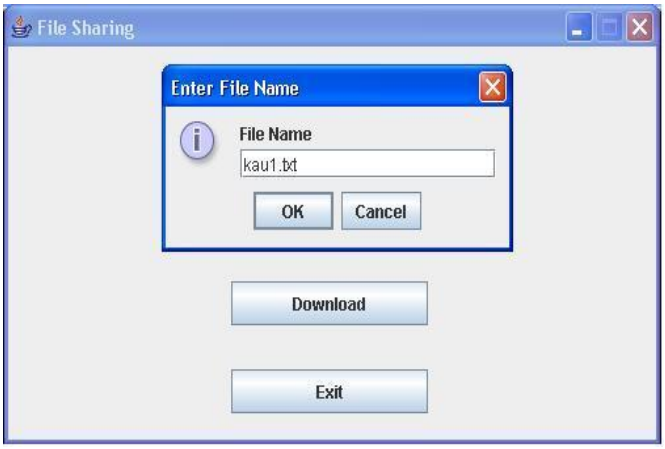

Fig 2: File Specified to the G-peer by querying

Using Java language as the front end and Microsoft SQL server 2005 as the back end, it is possible to call up a value (file) in the peers connected through a LAN (Local Area Network). The basic mechanism is to trigger the peers with search query. Here, in the Fig-2, a text file named "kau1.txt" is being asked by one of the peer in the network. If the file is not found in the G-peer-1, it shows the message "File not found on server". The variation of a peer's query reply action is demonstrated in Fig-3. The query is transmitted to the succeeding G-peer(n), with the message "Please communicate later... Server searching to locate the file on the other server…" Fig-4. This process helps us to find the file in any one of the G-peer with hit ratio of 90%. Finally the file can be retrieved from the G-peer (n-1). Fig-5.

Fig 3: Query to the next G-peer(s)

Fig 4: File searching by the G-peer

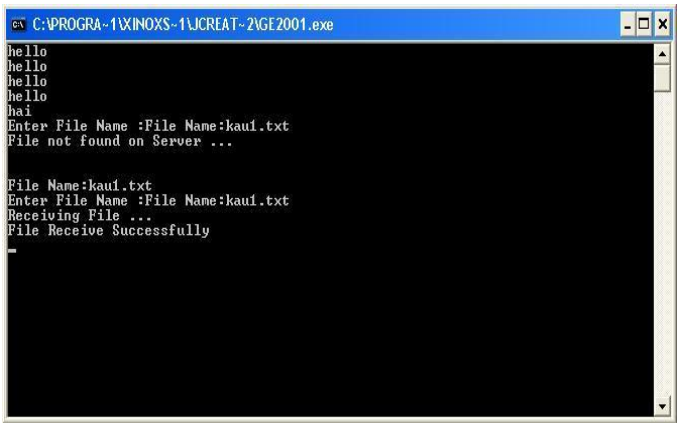

Fig 5: File queried is successfully retrieved

7. CONCLUSION

On Summarizing the contributions and limitations of my work, I discuss the directions for the future investigation. In my approach, the G-peers search algorithm proposed is a fully discrete approach which helps to increase the speed of file transfer. Also, the security implementation part in my authentication module serves to avoid the DDoS (Distributed Denial of Service) attacks [21].

8. ACKNOWLEDGEMENT

I would like to thank our editor, Kamaljit Lakhtaria for their valuable reviews and comments. Also, I thank my wife Revathi Saravanan and my kid Hariharasudhan for their support.

9. REFERENCES

[1] S. Saroiu, P. Gummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems", Proc. Multimedia Computing, 2002.

[2] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making Gnutella-Like P2P Systems Scalable," Proc. SIGCOMM, 2003.

[3] Q. Lv, S. Ratnasamy, and S. Shenker, "Can Heterogeneity Make Gnutella Scalable?" Proc. Int'l Workshop Peer-to-Peer Systems (IPTPS), 2002.

[4] E. Adar and B.A. Huberman, "Free Riding on Gnutella," http://www.firstmonday.dk/issues/issue5_10/adar/, 2003.

[5] B.Y. Zhao, L. Huang, J. Stribling, S.C. Rhea, A.D. Joseph, and J.Kubiatowicz, "Tapestry: A Resilient Global-Scale Overlay for Service Deployment," J. Selected Areas in Comm., 2004.

[6] R.M. Stoica, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. ACM SIGCOMM, 2001.

[7] B.F. Cooper, "An Optimal Overlay Topology for Routing Peer-to-Peer Searches," Proc. IFIP/ACM Middleware Conf., 2005.

[8] B.F. Cooper, "Quickly Routing Searches without Having to Move Content," Proc. Int'l Workshop Peer-to-Peer Systems (IPTPS), 2005.

[9] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," Proc. ACM SIGCOMM, 2001.

[10] MAYMOUNKOV, P., AND MAZIERES, D. Kademlia: A Peer-to-peer Information System Based On the XOR Metric. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02) (Cambridge, MA, Mar. 2002)

[11] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems," Proc. INFOCOM, 2003.

[12] S. Patro and Y.C. Hu, "Transparent Query Caching in Peer-to-Peer Overlay Networks," Proc. 17th Int'l Parallel and Distributed Processing Symp. (IPDPS), 2003.

[13] M. Roussopoulos and M. Baker, "Controlled Update Propagation in Peer-to-Peer Networks," Proc. USENIX Ann. Technical Conf., 2003.

[14] C. Gkantsidis, M. Mihail, and A. Saberi, "Random Walks in Peer-to-Peer Networks," Proc. INFOCOM, 2004.

[15] B. Yang and H. Garcia-Molina, "Efficient Search in Peer-to-Peer Networks," Proc. Int'l Conf. Distributed Computing Systems (ICDCS), 2002.

[16] B. Yang and H. Garcia-Molina, "Designing a Super-Peer Network," Proc. Int'l Conf. Data Eng. (ICDE), 2003.

[17] C. Wang, L. Xiao, and P. Zheng, "Differentiated Search in Hierarchical Peer to Peer Networks," Proc. Int'l Conf. Parallel Processing (ICPP), 2005.

[18] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hon ,"Freenet: A Distributed Anonymous Information Storage and Retrieval System", Marshall Aid Commemoration Commission and the National Science Foundation,2001.

[19] Nazareno Andrade, Miranda Mowbray, Aliandro Lim, Gustavo Wagne, Matei Ripeanu, "Influences on

Cooperation in BitTorrent Communities",SIGCOMM'05 Workshops, August 22–26, 2005, Philadelphia, PA, USA.

[20] Huu Tran,Michael Hitchens,Vijay Varadharajan,Paul Watters, "A Trust based Access Control Framework for P2P File-Sharing Systems",IEEE Proceedings of the 38th Hawaii International Conference on System Sciences - 2005.

[21] Felix Lau, Stuart H. Rubi, Michael H. Smith , Ljiljana Traj kovi, "Distributed Denial of Service Attacks", IEEE,2000.

10. AUTHORS PROFILE

Kaushik Sekaran received his Bachelor of Technology in Computer Science and Engineering from SASTRA University, Thanjavur in the year 2005. He then pursued postgraduate studies and received his Master of Engineering degree (in Computer Science and Engineering) from Mepco Schlenk engineering college, Affiliated to Anna Univesity, Chennai in the year 2008. Since March 2009 he is an Assistant Professor in School of computing science and Engineering at VIT University, Vellore, India. He is pursuing his Ph.D studies in VIT University. His current research interests include in the areas of distributed and Internet systems, overlay systems and applications, and Security issues in Peer-to-peer Networks. He is a member of IEEE Computer Society.