International Journal of Research Studies in Science, Engineering and Technology [IJRSSET]
Volume 1, Issue 2, May 2014, PP 17-22

The Deficit-based Distribution Algorithm in Bit Torrent Swarms for Peer-to-Peer File Sharing Allocation

S.Sundar raj
Research Scholar,
Department of Computer Science and Engineering,
HimalayanUniversity,
Arunachal Pradesh, India,
rajsundar1003@gmail.com

Dr. Ashish Chaturvedi
Professor and Associate Director,
Arni School of Computer Science and
Applications, Arni University, Indora (Kathgarh),
Himachal Pradesh,
India,dr_ashysh@yahoo.com

Abstract: *Peer-to-peer file-sharing claims suffer from a fundamental problem of unfairness. Free-riders cause slower download times for others by contributing little or no upload bandwidth while consuming much download bandwidth. Previous attempts to address this fair bandwidth allocation problem suffer from slow peer discovery, inaccurate predictions of neighboring peers' bandwidth allocations, underutilization of bandwidth, and complex parameter tuning. We present Fair Torrent, a new deficit-based distributed algorithm that accurately rewards peers in accordance with their contribution. A Fair Torrent peer simply uploads the next data block to a peer to whom it owes the most data as measured by a deficit counter. Fair Torrent is resilient to exploitation by free-riders and strategic peers, is simple to implement, requires no bandwidth over allocation, no prediction of peers' rates, no centralized control, and no parameter tuning. We implemented Fair Torrent in a Bit Torrent client without modifications to the Bit Torrent protocol and evaluated its performance against other widely used Bit Torrent clients. Our results show that Fair Torrent provides up to two orders of magnitude better fairness, up to five times better download times for contributing peers, better performance on average in live Bit Torrent swarms.*

Keywords: *BitTorrent, fairness, FairTorrent, peer-to-peer(P2P) networking, quality-of-service..*

1. Introduction

The Internet has witnessed a rapid growth in the popularity of various Peer-to-Peer (P2P) applications during recent years. In particular, today's P2P file-sharing applications (e.g., FastTrack, eDonkey,Gnutella) are extremely popular with millions of simultaneous clients and contribute a significant portion of the total Internet traffic . These applications have evolved over the past several years to accommodate growing numbers of participating peers. In these applications, participating peers form an overlay which provides connectivity among the peers, allowing users to search for desired files. Typically, these overlays are unstructured where peers select neighbors through a predominantly ad hoc process—this is different from structured overlays. Most modern file-sharing networks use

a two-tier topology where a subset of peers,Called ultrapeers, form an unstructured sparse graph while other participating peers, called leaf peers, are connected to the top-level overlay through one or multiple ultrapeers. More importantly, the overlay topology is continuously reshaped by both user-driven dynamics of peer participation as well as protocol-driven dynamics of neighbor selection. In a nutshell, as participating peers join and leave, they collectively, in a decentralized fashion, form an unstructured and dynamically changing overlay topology.

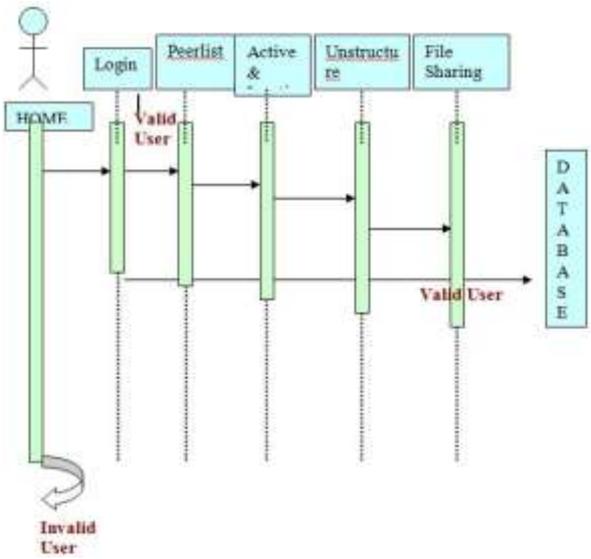

Fig1.1. *Sequence Diagram View of the Data Distribution*

This work focuses on developing an accurate understanding of the topological properties and dynamics of large-scale unstructured P2P networks, via a case study. Such an understanding is crucial for the development of P2P networks with superior features including better search, availability, reliability and robustness capabilities. For instance, the design and simulation-based evaluation of new search and replication techniques has received much attention in recent years. These studies often make certain assumptions about topological characteristics of P2P networks (e.g., a power-law degree distribution) and usually ignore the dynamic aspects of overlay topologies. However, little is known today about the topological characteristics of popular P2P file sharing applications, particularly about overlay dynamics. An important factor to note is that properties of unstructured overlay topologies cannot be easily derived from the neighbor selection mechanisms due to implementation heterogeneity and dynamic peer participation. Without a solid understanding of the topological characteristics of file-sharing applications, the actual performance of the proposed search and replication techniques in practice is unknown and cannot be meaningfully simulated. In this case study, we examine one of the most popular file-sharing systems, Gnutella, to cast light on the topological properties of peer-to-peer systems.

2. Literature Survey

BitTorrent [6] employs a rate-based TFT heuristic to incentivize peers to upload and attempt to provide fair exchange of bandwidth between peers. Peers participating in the download of the same target file form a swarm. The target file is conceptually broken up into pieces, typically 256 kB. Peers tell one another which pieces of the target file they already have and request missing pieces from one another. Requests are typically made for 16-kB subpieces. Peers that already have the entire file are seeds. Peers that are still downloading pieces of the file are leechers. TFT is used in a swarm to enable fair bandwidth exchange during the current download of a file. It operates by having each BitTorrent client upload to other peers in a round-robin fashion, where of the peers have provided the best download rate during the most recent 10–20-s time period, and peers are randomly selected to help discover other peers with similar upload rates. is typically between 4 and 10, and is typically 1 or 2. Thus, the active set of peers that a client uploads to is updated each round based on the measurements of their download rates. BitTorrent refers to the selection and deselection of a peer for uploading as unchoking and choking, respectively. Because of its popularity, much work has been done in studying BitTorrent's behavior. BitTorrent peers tend to exchange data with other peers with similar upload rates over a large file download [15]. Under some bandwidth distributions, the system has been shown to eventually converge to a Nash equilibrium [27]. However, there is no evidence that this behavior extends to shorter file downloads, dynamic environments, skewed distributions of users, or modified but compatible BitTorrent clients. In fact, several modified BitTorrent clients [20], [24], [30] have been developed that exploit different strategies to achieve better performance at the expense of users running unmodified BitTorrent. For example, BitTyrant [24] exploits the fact that BitTorrent will reciprocateat a higher rate even when receiving much smaller bandwidth from BitTyrant in return. The previous studies demonstrate that BitTorrent's TFT heuristic does not result in fair bandwidth exchange. Because TFT only identifies and exchanges data with a small number of peers at a time, a BitTorrent client may waste much time and bandwidth while discovering peers with similar upload rates in a large network [24]. Further waste occurs because relationships with discovered peers may be unstable, as the other peers are also always searching for better peers. Even after discovering peers with good upload rates, BitTorrent continues to blindly donate a portion of bandwidth by randomly uploading to other peers in hopes of reciprocation.

3. Methodology

3.1 Existing Method

Previous studies that captured P2P overlay topologies with a crawler either rely on slow crawlers, which inevitably lead to significantly distorted snapshots of the overlay , or capture only a portion of the overlay which is likely to be biased (and non-representative) . These studies do not examine the accuracy of their captured snapshots and only conduct limited analysis of the overlay topology.

3.2 Disadvantages

Unfortunately, More importantly ,these few studies are outdated (more than three years old), since P2P file sharing applications have significantly increased in size and incorporated several new topological features.

3.3 Proposed Method

Accurately capturing the overlay topology of a large scale P2P network is challenging. A common approach is to use a topology crawler that progressively queries peers to determine their neighbors. The captured topology is a snapshot of the system as a graph, with the peers represented as vertices and the connections as edges. However, capturing accurate snapshots.

3.4 Advantages

Overlay topologies change as the crawler operates

A non-negligible fraction of peers in each snapshot are not directly reachable by the crawler. When a crawler is slow relative to the rate of overlay change, the resulting snapshot will be significantly distorted.

Furthermore,verifying the accuracy of a crawler's snapshots is difficult due to the absence of authoritative reference snapshots. We introduce techniques for studying the accuracy of a crawler This work focuses on developing an accurate understanding

4. Peer-To-Peer File Sharing Allocation

Accurately capturing the overlay topology of a large scale P2P network is challenging. A common approach is to use a topology crawler that progressively queries peers to determine their neighbors. The captured topology is a snapshot of the system as a graph, with the peers represented as vertices and the connections as edges. However, capturing accurate snapshots is inherently difficult for two reasons:

(1)Overlay topologies change as the crawler operates and

(2) a non-negligible fraction of peers in each snapshot are not directly reachable by the crawler. When a crawler is slow relative to the rate of overlay change, the resulting snapshot will be significantly distorted. Furthermore, verifying the accuracy of a crawler's snapshots is difficult due to the absence of authoritative reference snapshots. We introduce techniques for studying the accuracy of a crawler This work focuses on developing an accurate understanding

List of Modules

- Authentication Module
- Peer List
- Active Peers and Inactive Peers
- Unstructured Topologies
- File Searching
- File Uploading
- File Downloading

Authentication Module:

The Authentication Module is main purpose of providing secure entry to project Phase. After the registration process are once registered by the User. The Authenticated user can enter into project.

Peer List:

The Main Purpose of searching a peer is to improve the communication between the two peers. In this Project, We are searching a peer list (gnutella network Method). According to our Paper Gnutella is a process to get more number of peers to improve file sharing process. Through this method we are getting peer list according to their company workgroup name

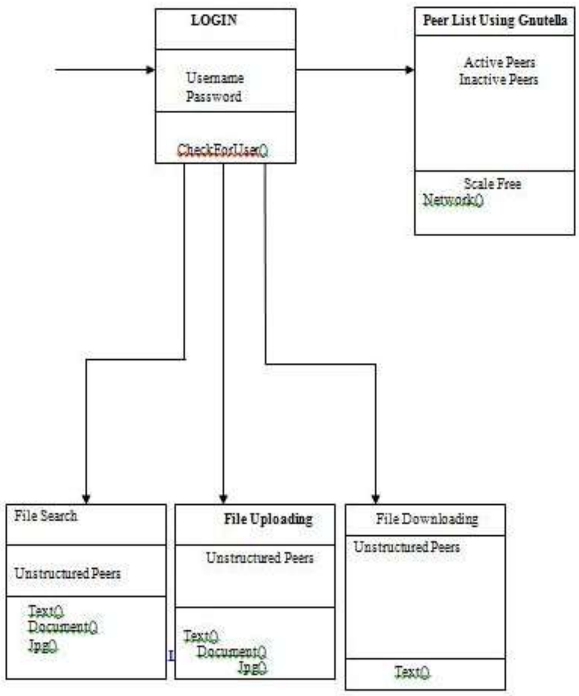

Fig 4.1 *Class View of File Allocation*

Active Peers and Inactive Peers:

After getting a peer list through gnutella network. We are Separating Active and Inactive peers. The System which is not responding, shutdown, improper network connection is said to be inactive peers. The System which are responding is said to be active peers

Unstructured Topologies:

In Unstructured topologies, these overlays are Unstructured where peers select

Neighbors through a predominantly Random process—this is different from Structure overlays

Process. In Our Project, We are generating Random Peer list to Communicate with each other Peers for file sharing process.

File Searching:

In this searching process, we are going to communicate with selected peers. We can search the files with active peers.

File Uploading:

In this file uploading Process, we are going to communicate with selected peers. We can Upload file to other peers. Through this method, we can upload all kind of files such as audio, video, images, documents

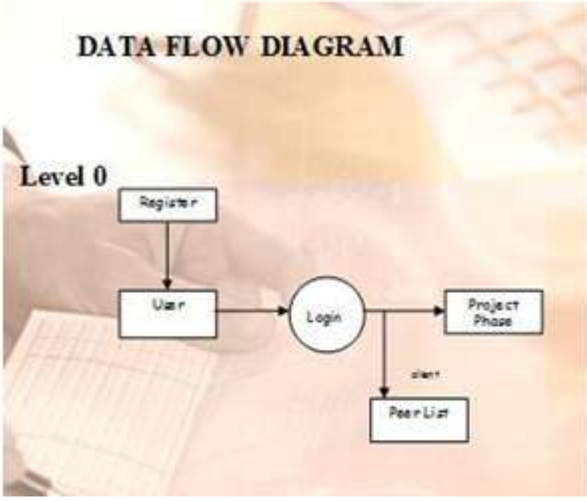

Fig 4.2 *Data Flow Diagram of Login in Data Sharing*

File Downloading:

In this file downloading Process, we are going to communicate with selected peers. We can Download files from other peers. Through this method, we can download and view all kind of files such as audio, video, images, documents.

5. Experiment Result

NET|| is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on). Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective. The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods. Implementation is the process of converting a new system design into operation. It is the phase that focuses on user training, site preparation and file conversion for installing a candidate system. The important factor that should be considered here is that the conversion should not disrupt the functioning of the organization. The application is implemented in the Internet Information Services 5.0 web server under the Windows 2000 Professional and accessed from various clients. An analysis of user training focuses on two factors

- User capabilities
- Nature of the system

Users range from the native to highly sophisticated. Hence they should be trained about the usage of software.The user should takes care to see that in the event of interruption due to power failure

SCREENSHOTS

Fig 5.1 *System Connection in the Work Group*

Fig 5.2 *Local Host Connection Establishment in Work Group*

Fig5.3. *Structure Topology for Source & Destination IP Address*

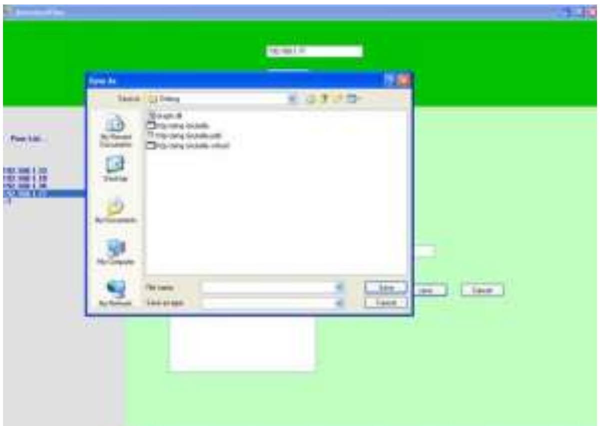

Fig5.4. *P2P Connection Establishment in the Peerlist*

Fig 5.5 *Active Peer Data Sharing for Bit Torrent Over the IP*

6. Conclusion

In this paper, we projected a general three-tier security framework for authentication and pairwise key establishment between mobile sinks and sensor nodes. The proposed scheme, based on the polynomial pool-based key redistribution scheme substantially improved network resilience to mobile sink replication attacks compared to the single polynomial pool-based key redistribution approach. Using two isolated key pools and having few stationary access nodes carrying polynomials from the mobile pool in the network may hinder an attacker from gathering sensor data, by deploying a replicated mobile sink. Analysis indicates that with 10 percent of the sensor nodes in the network carrying a polynomial from the mobile pool, for any mobile polynomial to be recovered, the attacker would have to capture multiple times more nodes as compared to the single polynomial pool approach.

We have further improved the security performance of the proposed scheme against stationary access node replication attack by strengthening the authentication mechanism between stationary access nodes and sensor nodes. We used the one-way hash chains algorithm in conjunction with the static polynomial pool-based scheme.

Reference

[1] I.F.Akyildiz,W.Su, Y. Sankarasubramaniam, and E.Cayirci, ―Wireless Sensor Networks: A Survey, ‖ Computer Networks, vol. 38, no. 4, pp. 393-422, 2002.

[2] T. Gao, D. Greenspan, M. Welesh, R.R. Juang, and A. Alm, ―Vital Signs Monitoring and patient Tracking over a Wireless Network, ‖ Proc. IEEE 27th Ann. Int'l Conf. Eng. Medicine and Biology Soc. (EMBS), Sept. 2005.

[3] L. Hu and D. Evans, ―Using Directional Antenna to Prevent Wormhole Attacks, ‖Proc. Network and Distributed System Security Symp., 2004.

[4] J.R. Douceur, ―The Sybil Attack, ‖ Proc. First Int'l Workshop Peer-to- Peer Systems (IPTPS '02), Mar. 2002.

[5] B.J. Culpepper and H.C. Tseng, ―Sinkhole Intrusion Indicators in DSR MANETs, ‖ Proc. First Int'l Conf. Broadband Networks (Broad- Nets '04), pp. 681-688, Oct. 2004.

[6] H. Deng, W. Li, and D.P. Agrawal, ―Routing Security in Wireless Ad Hoc Networks, ‖ Proc. IEEE Comm. Magazine, pp. 70-75, 2002.

[7] C. Intanagonwiwat, R. Govindan, and D. Estrin, ―Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, ‖ Proc. MobiCom, pp. 56-67, 2000.

[8] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin,―Intelligent Fluid Infrastructure for Embedded Networks, ‖ Proc. Second ACM Int'l Conf. Mobile Systems, Applications, and Services (MobiSys '04), June 2004.

[9] Y. Tirta, Z. Li, Y. Lu, and S. Bagchi, ―Efficient Collection of Sensor Data in Remote Fields Using Mobile Collectors, ‖ Proc. 13th Int'l Conf. Computer Comm. and Networks (ICCCN '04), Oct.2004.
[10] A. Rasheed and R. Mahapatra, ―An Energy-Efficient Hybrid Data Collection Scheme in Wireless Sensor Networks, ‖ Proc. Third Int'l Conf. Intelligent Sensors, Sensor Networks and Information Processing, 2007.
[11] W. Zhang, G. Cao, and T. La Porta, ―Data Dissemination with Ring-Based Index for Wireless Sensor Networks, ‖ Proc. IEEE Int'l Conf. Network Protocols (ICNP), pp. 305-314, Nov. 2003.
[12] B. Vijayakumar, Ashish Chaturvedi and K. Muthu Kumar, Effective Classification of Anaplastic Neoplasm in Huddling Stain Image by Fuzzy Clustering Method, International Journal of Scientific Research, Issue 3 volume 2, March-April 2013,
[13] H. Chan, A. Perrig, and D. Song, ―Random Key Pre-Distribution Schemes for Sensor Networks, ‖ Proc. IEEE Symp. Research in Security and Privacy, 2003.
[14] D. Liu, P. Ning, and R.Li. Establishing, ―Pairwise Keys in Distributed Sensor Networks, ‖ Proc. 10th ACM Conf. Computers and Comm. Security (CCS '03), pp. 52-61, Oct. 2003.
[15] H. Chan, A. Perrig, and D. Song, ―Key Distribution Techniques for Sensor Networks, ‖ Wireless Sensor Networks, pp. 277-303, Kluwer Academic, 2004.
[16] L. Eschenauer and V.D. Gligor, ―A Key-Management Scheme for Distributed Sensor Networks, ‖ Proc. ACM Conf. Computer Comm. Security (CCS '02), pp. 41-47, 2002.