International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) ISSN: 2395-5325
Volume 3, Special Issue 3 (September '2017)
Proceeding of - International Conference on Recent Trends in Computational Life Science and Information Technology,
Conference held at Tiruppur Kumaran College for Women,Tirupur,Tamilnadu, India.

TIER BASED SECURITY FOR ATTACKS IN BIT-TORRENT FILES

S.Selvi,
Assistant Professor,
Department of Computer Science,
Tiruppur Kumaran College for Women,
Tiruppur, Tamilnadu, India.

S.Nirmalajancy,
Research Scholar,
Department of Computer Science,
Tiruppur Kumaran College for Women,
Tiruppur, Tamilnadu,India.

.

Abstract: Bit-Torrent is one of most famous peer to peer file distribution server, that distribute large files in peer to peer manner such as Books, High Definition videos, Movies, albums, software's, Television serials, other than available in the Internet in the format of *.torrent files. It is also a Peer to Peer protocol, nowadays it is very popular software to receive the file from any server in the world. In Such case, it is world wide resource providing protocol through this peer to peer network, we can also download many number of useful files through internet. In all the networking services there have an intruder, in same way peer to peer have an high security threats nowadays. Such security threats are legally attacked, by many intruders through virus, worms, Trojans etc. Through this security threats many of them fear to use of Bit-Torrent protocol, through using of this software we defined the attack of threats to that protocol system users, so the user need to verify the downloaded content and discover if it is trusted or not. We need to stop the attackers who hamper the illegal file distribution in Bit-Torrent Protocol. In Existing research, they made Piece-Attack in Bit-Torrent protocol, in same way we need to prevent the attack and give the security of downloading to end users, the piece attack is an attack against the leeches in torrent networks, that is observed against attack has to be fully scaled. Not only the Piece attack, there having peer attack, fake-block attack, benevolent attack peers and also famous DDoS attack. Any attacks can to be prevent but in existing research they made only for piece-attack prevention. Proposed Research executes the prevention and secure based torrent files download from the peers, this is based on peer tier architecture. The architecture builds the peer in the manner when the receiver receives from the torrent files, the tier architecture is started from that leech peer and receiving peers are in the tier are made, from the tier the message is send to each and every peer to secure their content.

Keywords: Bit-Torrent, Piece Attack, Fake-Block Attack, Intruders

I.INTRODUCTION

Paradigm has verified to be a very valuable approach for designing scalable and tough networking applications. This approach beat the scalability limitations of the traditional client/server approach. Large scale file distribution and file sharing applications was the initial to exploit the new paradigm. Bit-Torrent, one of the most popular peer-to-peer (P2P) file distribution applications, was established by Bram Cohen in 2001. Bit-Torrent was quickly embraced by numerous content providers such as Windows, Blizzard and most Linux distributions as a scalable way of delivering content, decreasing the load on congested servers, minimizing the distribution cost and reducing the downloading time for users. It Torrent is a second generation P2P file distribution system, which focal point on organizing the peers that are involved in downloading the same file into an overlay network, called a Torrent. Interestingly, as more users link a Torrent, the downloading rate that is achieved by all the peers increases. Bit-Torrent is noted for introducing several innovative mechanisms including tit-for-tat (TFT) and rarest first (RF) that it uses for cooperative file sharing among the participating peers.

The widespread popularity of Bit-Torrent has attracted the attention of several researchers who conducted various performance studies in order to understand the behavior of the Bit-Torrent protocol, its mechanisms and the overall application performance. These studies affirmed the importance of TFT and rarest first mechanisms for Bit-Torrent's success. At the same time, several of these studies identified the impact of Bit-Torrent's phenomenal success on the traffic in computer networks. For example, one of the significant problems is the increase in file downloading traffic in the Internet. Reports show that Bit-Torrent has become the largest source of P2P file sharing traffic in the Internet: Bit-Torrent traffic accounted for 18% of broadband traffic in 2006. Karagiannis et al. also report that Bit-Torrent occupies about 13-15% of the access link bandwidth at a residential university. Furthermore, Bit-Torrent presents significant traffic-engineering challenges for Internet Service Providers (ISPs). Current implementations of Bit-Torrent ignore the underlying Internet topology or ISP link costs and result in a large amount of cross-ISP traffic. These issues show that Bit-Torrent protocol has further scope for improvement in its core mechanisms. For example, one of the most criticized of Bit-Torrent's mechanisms is the random neighbor selection strategy, which prolongs the file downloading time and results in inefficient usage of network resources. Some researchers also suggested Bit-Torrent-like

As stated earlier Bit-Torrent (BT) is one of the most popular applications for distributing large size files. The application is implemented as a hybrid P2P system, with most of the interaction directly among he peers, but requiring occasional interaction with a server for locating peers. The Bit-Torrent protocol enquires peers to organize themselves into an overlay network, with connections among the peers, for each file being distributed. This overlay network is called a Torrent. Each file being distributed by Bit-Torrent requires the establishment of separate torrent. Besides the peers, a tracker and a web server play an important part in file distribution using Bit-Torrent. The tracker is a special infrastructure node which stores meta-information about the peers that are currently active within a torrent. Peers interact with the tracker using a simple protocol layered on top of HTTP in which a peer sends information about the file it is

downloading and the port number to the tracker. The tracker does not participate in the actual distribution of the file, but only serves the purpose of enabling peers to find each other.

The peers that are part of a torrent can be classified into two types: a seed and a leecher. A seed is a client that has a complete copy of the file and remains in the torrent to serve other peers. For a torrent to get started, we need at least one initial seed that provides the entire content for download. A leecher is a client that is still downloading the file. When a file is to be made available for distribution through a torrent, then a special file with a .torrent extension is made available on a web server. The .torrent file contains information about the file including its length, name, hashing information, and the URL of a tracker. A peer that wants to download a file first obtains the corresponding .torrent file from the web server. Then, the peer contacts the tracker and requests a list of IP/port pairs of other peers that are already participating in the torrent. When the tracker receives a request, it will send the requesting peer a list of potential neighboring peers, usually 50, that are selected randomly from the set of all active peers within the torrent. After obtaining the list, the peer contacts around 20–40 peers from the list to add them as its neighbors. This set of neighbors forms the peer set for the joining peer. If the number of neighbors connected to a peer falls below 20, the peer will again contact the tracker

II.RELATED WORK

Taking into consideration Bit-Torrent as a two components function, the central tracker and the swarm, we can disagree that it has been the core study in many research works. Though, approximately all the studies on Bit-Torrent are focused on the swarm part and the tracker side but very few on Bit Torrents two DHTs, the Mainline DHT and the Azureus DHT. Among those, Wolchok et al. [21] conducted a monitoring study on the Azureus DHT. They clearly show how this DHT can be crawled thanks to a Sybil attack, so as to rebuild from scratch a Bit-Torrent search engine as well as to monitor pirate's behavior. While monitoring the Azureus DHT, the authors of [8] considered the performance of the lookup algorithm and propose new parameters for it reducing the lookup time at the cost of a moderate overhead.

They point out the difference of performances between the two DHTs despite the fact that they are both based on Kademlia and implement the same service. They also highlight many design and implementation problems affecting both the security and performance of these DHTs. Regarding the security of Bit-Torrent's DHTs, even fewer studies exist. In our previous work [19] we showed how the Mainline DHT network is wide open to attacks that can dangerously hurt the network, putting in jeopardy users privacy as well as the network performance itself. Recently, Jetter et al. proposed a self-registration mechanism, as a way to avoid a Sybil attack in Bit-Torrent DHTs. They limit the number of peers'ID per IP, so as to avoid an attacker to launch several peers from a single machine.

However, their solution require a jump to a new network, avoiding backward compatibility. Considering KAD, its performance has been first considered in [18] in which the authors highlight the role of K-bucketsin the efficiency and reliability of the routing table. Then [16] proposed a similar study but considered alternative parameters, like the number of contacts per lookup or the time window separating the lookup phase from the service.

Bit-Torrent is one of the most popular P2P protocols widely used by a huge number of Internet users all over the world. However, there are a few works about the attacks on Bit-Torrent networks. This is mainly because of its open design specification along with many popular open source client applications. In contrast, there are numerous amounts of research papers in P2P literature on both introducing and identifying diverse kinds of attacks against P2P networks which are orthogonal to the Bit-Torrent environment. As one of the early papers concerned with attacks against Bit-Torrent networks, [16] has discussed the Piece Lying and Eclipse attacks to hinder distribution of data in Bit-Torrent swarms. The authors evaluated the effectiveness of those attacks by launching them against their own implemented Bit-Torrent protocol, using a discrete-event simulator. They concluded that Bit-Torrent protocol is susceptible to those attacks and the targeted torrent networks can be taken down by attackers with even modest amounts of resources.

III.BACKGROUND STUDY

Problems In Present System

Bit Torrent peers are classify on each file basis as peers if they are downloading (and uploading) the file and seeders if they have the complete file and are uploading (only) to peers. A Trailer stores a small amount of state to help peers in discovering other peers. Files in Bit-Torrent consist of piece which in turn consists of blocks. A lecher l is involved in another peer p if p has pieces that l does not; likewise, l finds p attractive. All peers are involved in all seeders. Bit-Torrent peers may keep open links to a lot of neighbors, but usually only upload to, or un-choke, a small quantity of them. Users may adjust this number of unchoke slots, but it is generally either some small constant or some function of their up-load bandwidth. In previous work unchoking algorithm dictates to whom and how much to un-choke. To bootstrap, the un-choking algorithm consists of at least one arbitrary peer to hopefully un-choke regard-less of that peer's input. Peers notify one any more of the pieces they have; when they initial connected to a neighbor, they send a bit-array of their pieces, called a bit-field, which they later update with per-piece have messages. All peers maintain an approximation of each piece's availability: a count of how many neighbors have that piece. When a peer p begins un-choking leecher l, l informs p which of p's blocks it wishes to get next. The common approach is rarest-first, in which l prioritizes the pieces it views as least available.

Attack Survey In Bit-Torrent Based Peer To Peer Networks

1. Decoy Insertion

Decoy placing (or content pollution) is a method by which ruined versions of a picky file are inserted into the system. This deters users from result a virtuous version and also increases sharing of the corrupted file. A malicious user poison the file by change it into an additional format that is impossible to tell apart from uncorrupted files (example: it

may have similar or same metadata). In order to attract users to download the decoys, malicious users may make the corrupted file available via high bandwidth connections. This technique consumes a huge amount of computing resources since the malicious server must react to a large amount of requests. As a result, queries return first and foremost corrupted copies such as an empty file or executable records infected with a virus.

2. Index Poisoning

This method targets the directory found in P2P file sharing systems. The directory allows users to place the IP addresses of favored content. The Method of attack makes probing hard for network users. The assailant (attacker) inserts a large amount of invalid sequence into the directory to avoid users from finding the exact resource. Void information could include random content identifiers or forged IP addresses and port information. When a user effort to download the corrupted content, the server will fail to establish a connection due to the large quantity of invalid information. Users will then waste time annoying to establish a link with bogus users thus mounting the normal time it takes to download the file. The index poisoning attack requires less bandwidth and server resources than decoy insertion. Also, the assailant (attacker) does not have to transmit files nor respond to requests. For this reason, directory poisoning requires less effort than other methods of attack.

3. Spoofing

Some corporations that interrupt P2P file distribution on behalf of content contributors create their own software in order to launch attacks. Media protector has written their own program which directs users to non-existent locations via bogus search results. As users naturally choose one of the top five search results only, this technique requires users to persist beyond their initial failed efforts to locate the preferred file. The idea is that many users will simply give up their search through aggravation.

4. Interdiction

This technique of attack prevents distributors from allocation users and thus slows P2P file allocation. The attacker's servers always connect to the preferred file, which overflow the provider's upstream bandwidth and prevents other users from downloading the file.[6]

Selective Content Poisoning

Selective content poisoning (also known as proactive or discriminatory content poisoning) attempts to detect copyright violators while allowing legitimate users to continue to enjoy the service provided by an open P2P network. The protocol recognizes a peer with its endpoint address even as the file index set-up is altered to incorporate a digital signature. A peer confirmation protocol can then establish the authority of a peer when she downloads and uploads files. Using identity based signatures, the system allow each peer to recognize infringing users without the requirement for communication with a central authority.

The protocol then sends poisoned chunks to these notice users requesting a copyright sheltered file only. If all rightful users just deny download requests from recognized infringers, the latter can regularly accumulate clean chunks from colluders (paid peers who distribute content with others without approval). Though, this technique of content poisoning forces illegal users to discard even clean chunks, prolonging their download time. Voluntary communal Licensing and the Open Music Model are theoretical systems where users give a subscription charge for access to a file distribution network, and are able to officially download and distribute copyright content. Choosy content poisoning could potentially be used here to edge access to legal and subscribed users, by given that poisoned content to non-subscribed users who attempt to illegally use the network.

Eclipse Attack

The eclipse attack (also known as routing-table poisoning), as a substitute of poisoning the network, aim requesting peers directly. In this attack, the aggressor (attacker) takes in excess of the peer's routing table so that they are unable to communicate with any other peer excluding the attacker. As the aggressor repeat the whole network for the aimed peer, they can manipulate them in a number of ways. For example, the attacker can denote which search results are returned. The attacker can also alter file explanation. The peer's requirements can also be directed back into the network by the attacker and can also be customized. It also verify data randomly for any errors found in that.

Uncooperative-Peer Attack

In this attack, the attacker links the targeted swarm and begins connections with many peers. Though, the attacker never offers any chunks (authentic or if not) to the peers. A common version of this attack is the "chatty peer" attack. The attacker found connection with targeted peers via the required handshake message, followed by message advertising that they have a number of existing chunks. Not only does the attacker never offer any chunks, they also frequently resend the handshake and message. These attacks avoid downloads as, basically, the peer wastes time dealing with the aggressor, as a alternate of downloading chunks from others.

Piece Attack

In Piece-Attack, the attackers try to fail the hash verification phase of the victim user by uploading at least one fake block to it. When the victim downloads all the blocks from different peers, the victim client application concatenates the blocks and then calculates the SHA1 hash value of the entire piece. When there is at least one fake block within the piece, the hash verification will fail. Since it is not possible to identify the fake block, the victim has to download all the blocks within the same piece again. However, as a simple defense mechanism, the victim can put all the peers which provided the fake piece in a black list and then start downloading the piece from neighbors outside that list again. This defense mechanism is effective only when we have sufficient number of innocent active peers in the swarm. Because, it takes some times to ask the tracker for introducing new peers, as mentioned before. To calculate the efficiency of the Piece-Attack, we give the following example: suppose we have an 8 GB medium size torrent with 512 KB pieces (typically each of them has 32 blocks).

The attacker can waste an entire 512 KB download of the victim, by only sending a 16 KB block (less than 7%). It is necessary to note that this is not a practical approach to avoid this attack by reducing piece length. This is because each piece occupies at least 20 Bytes for its hash value within the torrent metadata file and given that, too many small pieces will cause huge torrent metadata files

IV.PROPOSED METHODOLOGY

Proposed Research

To allocate a file, the Bit-Torrent client breaks the file into several pieces of equal size according to a "meta-info" file. Members of the swarm announce which pieces they have and which pieces they want. Pieces are further divided into 16 kB "blocks" to facilitate distribution. Bram Cohen's original design anticipated that the size of a piece would be 250 kilobytes, but observation demonstrates variance in the wild, particularly for small torrents and torrents over 10 Gb.

A piece attack, except that multiple Sybils act in a coordinated manner to control the target peer's perception of the swarm, preventing it from receiving all pieces of the torrent. This manipulation is typically accomplished by falsifying the bit-field messages and referring the target to seek pieces only from other peers participating in the attack. However, the targeted peer will also receive a randomly generated peer list from the tracker. Therefore, the greater the number of coordinated peers in the swarm as a percentage of the whole, the more likely the peer list will contain only the attacking peers. In addition to refusing to share, coordinated peers can reinitiate the handshake sequence multiple times, in what the literature refers to as a "chatty peer attack".

In such a way the proposed technique introduce the falsifying the bit-field message we given through Reactive measurement methodology While repeatedly downloading a file suspected to be under attack, we collected multiple packet traces from hosts connected to both Ethernet and DSL access net-works. For this testing, we used Azureus and uTor-rent, as they are the two most widely used Bit-Torrent clients. On each host, we ran Wireshark (or TCP-dump) to capture all the incoming and outgoing packets. We also developed our own packet parser to identify different types of attackers in the trace and analyze their behaviors.

We evaluate the accuracy of Attack I by introducing the following three Tiers

$$P1=\frac{|U \cap RT|}{|U|}$$

$$P2=\frac{|C_{Peers} \cap RT|}{|C_{Peers}|}$$

$$P3=\frac{|N_{Peers} \cap RT}{|N_{Peers}|}$$

where U is a set of all shortened PEERSs posted by followings of the target user, Cpeerss is a candidate PEERS set including shortened PEERSs inferred as visited by the target user, Npeerss is a non-candidate PEERS set including shortened PEERSs nferred as unvisited, and RT is a retweeted PEERS set includ-ing shortened PEERSs which are in the target user's timeline or favorite lists, satisfying the following conditions:

$$C_{Peers} \subseteq U, N_{Peers} \subseteq U, RT \subseteq U,$$
$$C_{Peers} \cup N_{Peers} = U, C_{Peers} \cap N_{Peers} = \phi.$$

Therefore, P1 indicates the fraction of retweeted shortened PEERSs, P2 indicates the fraction of retweeted candidate PEERSs, and P3 indicates retweeted non-candidate PEERSs. The final values of the three metrics are as follows: P1 is 0.032, P2 is 0.048, and P3 is 0.003. P2 is 1.5 times higher than P1 and 16 times higher than P3, which implies that we can successfully categorize all shortened PEERSs into a set of visited PEERSs and a set of unvisited PEERSs. The target users normally post tweets containing shortened PEERSs included in the candidate PEERSs set and rarely post tweets with shortened PEERSs outside the candidate PEERSs set. According to boyd et al. approximately three percent of tweets are likely to be retweeted. This percentage is similar to our calculation of P1, 0.032; therefore, we conclude that the value of P1 is trustworthy. Next, we introduce two different metrics to view the Results from a different angle:

$$P_4=\frac{|C_{Peers} \cap RT}{|RT|}$$

$$P_5=\frac{|N_{Peers} \cap RT}{|RT|}$$

P4 indicates the fraction of retweeted PEERSs included in the candidate PEERS set and P5 indicates the fraction of retweeted PEERSs included in the non-candidate PEERS set. The final values of the two metrics are as follows: P4 is 0.952 and P5 is 0.048. We observe that P4 is much higher than P5, which implies that most of the shortened PEERSs that are in the timeline or favorites of the target users are inferred as candidate PEERSs. Therefore, we conclude that a shortened PEERS is highly likely to be retweeted or favorited by the target user if it is included in the candidate set. The problem of the metrics P4 and P5 is that we can simply achieve the best results when we include all the shortened PEERSs in the candidate PEERS set. To show that we correctly infer the candidate PEERS set, we compute the reduction ratio RR, which represents how much we reduce the number of candidate PEERSs from the number of all shortened PEERSs posted by the followings of the target user. RR is computed as follows:

$$RR=\frac{|C_{Peers}|}{|U|}$$

The reduction ratios of shortened PEERSs from the two PEERS shortening services: the average value of the reduction ratio is 0.669. Since the reduction ratio is quite smaller than P4, we can conclude that our attack system intelligently reduces the candidate set.

V.CONCLUSION

This research presented an Attack based of Bit-Torrent networks. First we evaluate the presentation studies of the original Bit-Torrent's protocols with measurement, simulation, and systematic modeling. We then summarized the findings of these studies. Next, we summarize some of the suggested development to Bit-Torrent's mechanisms in order to further progress its performance. As Bit-Torrent has become one of the most admired peer-to-peer content distribution protocols, various security threats have appeared from its vulnerable protocol planning. In this research, we analyze the vulnerabilities of Bit-Torrent protocol, and review present attacks on the protocol. From our research result that we prevent bit-torrent piece attack and other attacks.

VI.REFERENCES

[1]. Ipoque, "Internet Study 2008/2009," available at http://www.ipoque.com
[2]. B. Cohen, "The Bit-Torrent Protocol Specification," Available: http://www.Bit-Torrent.org/beps/ bep_0003.html
[3]. K. C. Sia, "DDoS vulnerability analysis of Bit-Torrent protocol," University of California, Los Angeles, USA, 2007.
[4]. [4] A. Loewenstern, "DHT Protocol," Available: http://www.Bit-Torrent.org/beps/bep_0005.html
[5]. K. E. Defrawy, M. Gjoka, and A. Markopoulou, "BotTorrent: misusing Bit-Torrent to launch DDoS attacks," USENIX'2007, Jun. 2007.
[6]. J. Harrington, C. Kuwanoe, and C. C. Zou, "A Bit-Torrent-driven distributed denial-of-service attack," SecureComm'2007, Sep. 2007.
[7]. X. Sun, R. Torres, and S. Rao, "Preventing DDoS attacks on internet servers exploiting P2P systems," Computer Networks, Elsevier North-Holland, Oct. 2010.
[8]. M. A. Konrath, M. P. Barcellos, and R. B. Mansilha, "Attacking a Swarm with a Band of Liars: evaluating the impact of attacks on Bit-Torrent," P2P'2007, Sep. 2007.
[9]. A. Hegenberg, "Attacks and exploits targeting Bit-Torrent and other P2P file sharing networks," FI'2009, Nov. 2009.
[10]. [10] P. Dhungel, D. Wu, B. Schonhorst, and K. W. Ross, "A measurement study of attacks on Bit-Torrent leechers," IPTPS'2008, Feb. 2008.
[11]. S. Majing, Z. Hongli, F. Bingxing, and D. Xiaojiang, "DDoS vulnerability of Bit-Torrent Peer Exchange extension: Analysis and defense," ICC'2012, Jun. 2012.
[12]. P. Dhungel, et al., "A Measurement Study of Attacks on Bit-Torrent Seeds," ICC'2011, Jun. 2011.
[13]. B. S. Sarjaz and M. Abbaspour, "Securing Bit-Torrent using a new reputation-based trust management system," Peer-to-Peer Networking and Applications, Springer US, Mar. 2013.
[14]. L. Wang and J. Kangasharju, "Real-world sybil attacks in Bit-Torrent mainline DHT," GLOBECOM'2012, Dec. 2012.
[15]. F. Pontes, F. Brasileiro, and N. Andrade, "Bit-Torrent Needs Psychiatric Guarantees: Quantifying How Vulnerable Bit-Torrent Swarms are to Sybil Attacks," LADC'2009, Sep. 2009.
[16]. M. Engle and J. I. Khan, "Vulnerabilities of p2p systems and a critical look at their solutions," Technical Report, Kent State University, Nov. 2006.
[17]. M. Vestola, "Security Issues in Structured P2P Overlay Networks," TKK Technical Reports in Computer Science and Engineering, Aalto University, 2010.
[18]. Y. Yang, et al., "A Survey of Peer-to-Peer Attacks and Counter Attacks," SAM'2012, Jul. 2012.
[19]. P. Dhungel, X. Hei, K. W. Ross, and N. Saxena, "The pollution attack in P2P live video streaming: measurement results and defenses," Sigcomm'2007, Aug. 2007.
[20]. J. P. Timpanaro, T. Cholez, I. Chrisment, and O. Festor, "Bit-Torrent's Mainline DHT Security Assessment," NTMS'2011, Feb. 2011.
[21]. P. Dhungel, D. Wu, and K. W. Ross, "Measurement and mitigation of Bit-Torrent leecher attacks," Computer Communications, Elsevier, Nov. 2009.
[22]. M. P. Barcellos, et. al., "Protecting Bit-Torrent: Design and Evaluation of Effective Countermeasures against DoS Attacks," SRDS'2008, Oct. 2008.
[23]. J. K. So, "Defending Against Malicious Behaviors in Bit-Torrent Systems," North Carolina State University, 2012.
[24]. B. Giovanni, "A Distributed Denial-of-Service (DDoS) Attack using Bit-Torrent Peer-to-Peer (P2P) Network," Internet Sicherheit (Seminar), Technische University, 2008.
[25]. S. Rouibia, J. Vayn, O. Beauvais, and G. Urvoy-Keller, "Early Stage Denial of Service Attacks in Bit-Torrent: An Experimental Study," WETICE'2008. Jun. 2008.
[26]. B. Cohen, "Incentives build robustness in Bit-Torrent," IPTPS'2003, Feb. 2003.
[27]. V. Atlidakis, M. Roussopoulos, and A. Delis, "Changing the Unchoking Policy for an Enhanced Bit-Torrent," Parallel Processing, Springer Berlin Heidelberg, 2012.
[28]. M. Slot, "Latency-driven Bit-Torrent," Master thesis, Vrije University, Aug. 2008.
[29]. R. Binbin, X. Wei, C. Hao, and Y. Dejian, "Improving Locality of Bit-Torrent with ISP Cooperation," ICECT'2009, Feb. 2009.
[30]. V. Pacifici, F. Lehrieder, and G. Dan, "Cache capacity allocation for Bit-Torrent-like systems to minimize inter-ISP traffic," INFOCOM'2012, Mar. 2012.