A Large-Scale Active Measurement Study on the Effectiveness of Piece-Attack on BitTorrent Networks

Ali Fattaholmanan, *Member, IEEE* and Hamid R. Rabiee, *Senior Member, IEEE*

Abstract—The peer to peer (P2P) file sharing applications have allocated a significant amount of today's Internet traffic. Among various P2P file sharing protocols, BitTorrent is the most common and popular one that attracts monthly a quarter of a billion users from all over the world. Similar to other P2P file sharing protocols, BitTorrent is mostly used for illegal sharing of copyright protected files such as movies, music and TV series. To impede this huge amount of illegal file distributions, anti-P2P companies have arisen to stand against these applications (specially the BitTorrent). To this end, they have begun to fire large-scale Internet attacks against BitTorrent networks. In this paper, we are going to actively measure the impact of the piece-attack against BitTorrent networks. Our measurement is divided into five scenarios in order to figure out the constraint factors that influence the success of the attack. To be able to evaluate the attack in different experiments, we defined *attack effectiveness* to quantitatively verify the success of the attack. Based on the measurement results, we discovered how it is possible to achieve significant outcome with modest amount of resources used by the attackers in hampering the illegal distribution of files in BitTorrent networks.

Index Terms—Peer-to-peer, BitTorrent network, piece-attack, measurement study, internet attack

1 Introduction

In recent years, peer to peer (P2P) applications and protocols have been widely spread all over the world and gained a considerable popularity among Internet users. As stated in [1], about 25 percent of overall Internet bandwidth is allocated to the P2P traffic. Among all the P2P protocols, BitTorrent is the most well-known protocol [2], which is widely used for sharing large files such as movies, music and TV series. Currently, BitTorrent has 150 million concurrent active users and about a quarter of a billion users monthly [3]. This considerable amount of users contributes to more than 17 percent of overall Internet bandwidth [1] which obviously reveals the outstanding features of this P2P file sharing protocol.

BitTorrent either can provide an inexpensive and scalable technique for file distribution, as used by some not-for-profit software corporations (e.g., Eclipse [4] and Linux [5]), or can be used for downloading copyright protected files, illegitimately. Since BitTorrent protocol and its client applications were not designed and developed by a single corporation, it is impossible to settle a lawsuit against them. Moreover, in most popular BitTorrent clients such as uTorrent, Vuze (Azureus) and FlashGet, *Peer Discovery* can be handled in a distributed manner [6] without the existence of any centralized entity (i.e., tracker) which makes it even harder for copyright enforcement agencies to hamper BitTorrent lawfully. Unfortunately, nearly two-thirds of current BitTorrent traffic belongs to illegal sharing of copyright protected files [1] such as music, movies or software. Consequently, movies and music industries have started to hire anti-P2P companies [7] to impede the distribution of targeted music, movies and other products protected by copyright over P2P file sharing networks (i.e., BitTorrent). Those anti-P2P companies are attempting to alleviate the illegal distribution of copyright protected products using two different techniques:

1) *Monitoring BitTorrent networks*. As stated in [8], there are some agencies (e.g., Media Defender [9]), which consequently monitor BitTorrent networks, especially networks with popular contents. By monitoring, they can send digital millennium copyright act (DMCA) takedown notice to the end-users contributing to sharing of copyright protected materials. As an evidence of the activeness of this technique, it is worth noting that most of the US universities have established rules about DMCA takedown notification received by college students (e.g., [10], [11], [12]). This is because of the increasing demand for illegal music downloading among US college students [13]. Unfortunately, it is possible to easily bypass the monitoring agencies without worrying about DMCA takedown notifications. For instance, as stated in [8], there are some available IP block lists in order to preserve BitTorrent end-users from establishing connection to anti-P2P companies (e.g., Media Defender) or government related domains (e.g., DoD). In addition, many copyright holder agencies currently use inconclusive methods for identifying BitTorrent end-users contributing to illegal distribution of copyright protected files. The authors in [14] demonstrated

- *The authors are with the AICT Innovation Center, Department of Computer Engineering, Sharif University of Technology (SUT), Tehran 145888, Iran. E-mail: fattah@ce.sharif.edu, rabiee@sharif.edu.*

Manuscript received 1 Apr. 2014; revised 20 Nov. 2014; accepted 25 Mar. 2015. Date of publication 6 Apr. 2015; date of current version 2 Sept. 2016. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TDSC.2015.2420561

a simple practical technique for implicating innocent end-users in illegal content sharing.

2) *Internet attack against BitTorrent networks*. Since *Monitoring BitTorrent Networks* cannot successfully stop end-users from downloading copyright protected content illegally, anti-P2P companies went beyond just monitoring BitTorrent networks and attempted to begin attacks against them. There are various kinds of attacks against BitTorrent networks based on the victim entity [7] (such as attacks on leechers, seeders, peer discovery and torrent discovery). In [7], it was observed that the BitTorrent networks of top popular movies are under various kinds of attacks including Piece-Attack and Connection-Attack. However, according to the significant proportion of illegally traffic allocated by BitTorrent end-users, their results are not promising. Here, a question arises: “How can we get more impressive results from those attacks?” and consecutively “How much resources and equipment is necessary to have such a worthy outcome?”

In this paper, we actively measure the effectiveness of Piece-Attack on BitTorrent networks. Piece-Attack is one of the attacks against leechers in BitTorrent networks that was first observed against real torrent swarms in [7]. However the effectiveness of the attack has not been actively measured yet. The contributions of this paper include:

- We actively measure the effectiveness of Piece-Attack by launching it against different kind of real BitTorrent networks. We have fired large-scale Piece-Attacks, via numerous public IP addresses used by hundreds of attacker peers. We repeated our measurements in several *Scenarios* to see the results of the attack against different kinds of BitTorrent networks.
- We point out the constraint factors that anti-P2P companies should consider in using this kind of attack against peers who contribute to public distribution of copyright protected materials in BitTorrent networks. To accurately measure the factors that can affect the intensity of Piece-Attack, in each scenario, we have fired lots of attacks with variant number of public IP addresses used by our attacker peers and also diverse number of attackers. During these attack scenarios, we measured the amount of resources reserved by the attack to estimate the cost and the amount of resources needed for anti-P2P companies in order to considerably harass users who are downloading illegally from BitTorrent networks.
- We show how anti-P2P companies can achieve notable results with a few resources using this attack. To this end, we have defined *attack effectiveness (AE)*, indicating how longer an arbitrary victim peer should linger in order to download a torrent file, completely. *Attack effectiveness* is a useful factor for evaluating the success of the attack. We calculate it for each measurement to quantitatively determine the attack successfulness. We have also provided an analytical model in order to predict it with regard to the amount of resources allocated by the attackers. The model also can evaluate the amount of bandwidth required for the attack. In other words, our model helps anti-P2P companies to calculate the attack’s cost (i.e., number of attackers, number of public IP addresses and the amount of required data bandwidth) for a specific result.
- In order to launch numerous numbers of attackers against a targeted BitTorrent network, we designed and developed an innovative Semi-Nat [15] protocol in order to provide multiple IP addresses for torrent client applications, running on our attack host.

The rest of this paper is organized as follows: Section 2 discusses related work, while Section 3 briefly introduces the Piece-Attack and gives a simple model on the effectiveness and cost of the attack on BitTorrent networks. Section 4 addresses our large-scale active measurement scenarios, which was intended to evaluate the impact of Piece-Attack on various types of BitTorrent networks. Finally, Section 5 draws conclusions.

2 Related Work

BitTorrent 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 BitTorrent 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 BitTorrent environment.

As one of the early papers concerned with attacks against BitTorrent networks, [16] has discussed the *Piece Lying* and *Eclipse* attacks to hinder distribution of data in BitTorrent swarms. The authors evaluated the effectiveness of those attacks by launching them against their own implemented BitTorrent protocol, using a discrete-event simulator. They concluded that BitTorrent protocol is susceptible to those attacks and the targeted torrent networks can be taken down by attackers with even modest amounts of resources. They assumed an identical behavior for all the BitTorrent clients in their simulation which is not practically true in real torrent swarms. Today, there are different kinds of BitTorrent client applications that are well configured to get torrent files with high download rates and minimum amount of data uploading. To achieve this goal, those client applications do not necessarily obey the standard BitTorrent protocol. Thus, it is not possible to accurately model the behavior of real-world BitTorrent networks.

The authors in [7], a one of the rare measurement studies with simulation on this literature, have observed *Piece-Attack* and *Connection Attack* fired by anti-P2P companies against eight top box-office movies torrent swarms. According to their passive measurement on leecher attacks, they have figured out that those attacks can prolong the average download time to more than twice the normal time. However, this may not considerably affect the satisfaction of BitTorrent users and accordingly, cannot stop the illegal distribution of copyright protected content in BitTorrent networks. They have also introduced and simulated some heuristic solutions as a defense mechanism against Piece-Attack for BitTorrent users. Moreover, they have only measured the current state of the leecher attacks on BitTorrent networks, but they did not fire any attack by themselves.

An interesting work on actively evaluating the effectiveness of a specific attack on P2P networks was expressed in [17], where a single attacker was capable of severely compromising a real live P2P streaming system. They verified the impact of the *Pollution-Attack* by monitoring the life time of the peers contributing the targeted P2P network. They observed a sharp decrease on the average life time of the peers, right after the start of the emphPollution Attack. However, this attack is not practical in P2P file sharing networks such as BitTorrent, because of the hash checking techniques used by these protocols.

To the best of our knowledge, [18] is the only work that tries to actively measure the effects of *Fake-Block Attack* (another name for Piece-Attack) against BitTorrent networks. They have stated that *Fake-Block Attack* is capable of prolonging the download time of a victim peer, from 2 to 11 times more than downloading in a safe network that is not under attack. However, they did not mention the amount of resources reserved for their attack. These resources include the number of attackers, number of public IP addresses they used and number of innocent leechers and seeders contributing to the targeted torrent swarm. This is in contrast to our work where we find the relationship between the amount of resources allocated by the attackers and the effectiveness of the Piece-Attack. Moreover, we figure out, how we can get the most promising result.

In this paper, we are going to assess how vulnerable are real BitTorrent networks to the Piece-Attack by deploying the attack against real BitTorrent networks. To measure the success of the attack, we run some torrent clients during the attack to observe how longer an arbitrary victim peer should linger in order to completely download a torrent file. Moreover, we measured the correlation between the impact of the attack and various parameters such as number of attacker clients, number of public IP addresses used by them, number of innocent seeders contributing to the torrent, and the age of the targeted swarm.

3 Preliminary Insight into the Piece Attack

In order to introduce our model for Piece-Attack and how it is supposed to calculate the *attack effectiveness*, we provide a brief background on both the BitTorrent protocol itself and how Piece-Attack is going to exploit its vulnerabilities. Thus, in the remaining of this section, we first explain the basic concepts of the BitTorrent protocol and then we will describe the techniques used in our experiment to fire Piece-Attack against the BitTorrent protocol in real-world circumstances. Finally, we introduce a simple and comprehensive analytical model for predicting the *attack effectiveness*, a criterion to accurately evaluate the attack successfulness. Finally, the model is extended in order to calculate the amount of bandwidth consumed during the attack.

3.1 BitTorrent

To download a file using BitTorrent protocol, first the user should download a `.torrent` metadata file which contains the list of files contained in the torrent with their sizes, piece length, SHA1 hash values of each piece, and the URL address of at least one tracker. Torrent metadata file is usually less than 50 KB in size and can easily be obtained from the Internet (e.g., torrent engine websites). Each torrent file is divided into several pieces with a fixed length, usually power of 2, as is specified in the torrent metadata file [19].

After downloading the `.torrent` metadata file, the client application reads the file and starts the *Peer Discovery* process in which it tries to connect to the trackers for getting a random set of active peers already connected to the swarm. To obtain the list of pieces, the client starts to establish a connection to each peer. The client will then send some piece requests in order to download the pieces from them. To achieve a higher download rate, the client divides each piece into several blocks, often 16 KB blocks [19], and then downloads blocks from its neighboring peers simultaneously.

After downloading all the blocks, they form a single piece. While a piece is downloaded completely, the client will verify its content by calculating the SHA1 hash value of the piece and comparing it with the value that already was stated in the torrent metadata file. If the hash values were the same, the piece is fine and the client keeps the piece and updates the tracker about its status and the pieces it has. This will help the tracker to recommend the client to other leechers requesting new peers. Commonly, the peers who have downloaded all the pieces are named *seeders* while other peers are called *leechers*.

During the download time, each peer periodically requests new peers from the tracker to find those with higher upload capacity. Then, each peer in the swarm tries to increase its download rate by regularly looking for new neighbors with higher upload rate. To limit the number of peer requests coming from clients, each tracker specifies a wait time interval to ask clients to wait between their regular requests.

3.2 Piece-Attack Specification

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 percent). 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 [19].

3.3 Simple Analytical Model

In this section, we present a simple yet accurate analytical model to help us precisely predict the effectiveness of the Piece-Attack on BitTorrent networks. In this model, we calculate the *attack effectiveness*, which is defined as the expected extra time that a victim should wait until completely downloading a targeted torrent file when it is under attack. To develop our model, we should face hardness of the simplicity vs. accuracy tradeoff in real networks. To this end, we first assume that every single attacker has a unique public IP address during the attack. This is because of the default option in almost any BitTorrent client application that does not accept connections from two different peers with identical IP address. We also assume that victim clients do not ban connections from the peers that were cooperating in uploading the pieces that failed in the hash verification phase. This assumption helps us to achieve a tractable model. However, as we discussed earlier, it can be a very simple but efficient defensive technique against the Piece-Attack.

According to the above assumptions, we develop our model based on the work in [7]. We extend that model to evaluate the *attack effectiveness* for an ordinary victim in the torrent network. Let *n* be the number of victim's neighbors that claim to have a specific piece. Among those *n* peers, *m* peers are attackers and consequently, we have *n*−*m* arbitrary innocent peers within the neighbors. Let *k* denote the average number of unique peers that a normal client will connect to, for downloading all the blocks of that specific piece. In our previous example, *k* is a number between 1 and 32, however, *k* = *10* is a common value for a client with a 5 Mbps link and 50 ms delay [19]. The victim will be affected from the attack if there was at least one attacker among those *k* peers. Thus, the probability of cleanly downloading a piece, is given by:

$$\begin{aligned}\Gamma &= P(\text{download a piece cleanly}) = \frac{\binom{n-m}{k}}{\binom{n}{k}}\\ &= \frac{n}{n-k}\times\frac{n-1}{n-k-1}\times\cdots\times\frac{n-m+1}{n-k+1}\\ &\approx \left(1-\frac{m}{n}\right)^k,\end{aligned} \qquad (1)$$

where $k \ll n, n-m$. However, in the steady state, and also by assuming that the neighbors are chosen from the swarm uniformly (as mentioned in [19]), we can replace $\frac{m}{n}$ with $\frac{M\times Q}{N}$ where the symbol *N* is the total number of active peers in the swarm, *M* is the total number of attacker peers, and *Q* is defined as follows:

$$Q = \frac{\text{average number of neighbors our attackers have}}{\text{average number of neighbors an ordinary peer has}}$$

Thus:

$$\Gamma \approx \left(1-\frac{M\times Q}{N}\right)^k. \qquad (2)$$

The attacker can specify M and Q in Eq. (2) and obtain the value of Γ (by estimating the variable N and assuming a constant value for k).

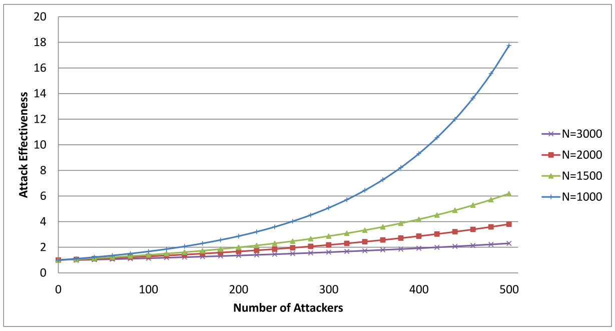

Fig. 1. *Attack effectiveness* as a function of M (number of attackers) for various number of N (swarm sizes). We used $k = 10$ and $Q = 0.5$ in the proposed model.

Now, we want to evaluate the *attack effectiveness*; the expected extra time an ordinary victim should wait until completely downloading a targeted torrent file during the attack. As mentioned before, a victim peer will retry downloading a piece until the downloaded data passes the hash verification phase. To evaluate the *attack effectiveness*, we must compute the expected number of times the victim downloads a piece uncleanly, in order to have the flawless one.

To this end, let us define $Pr(X = i)$, the probability of downloading a specific piece cleanly, exactly in the ith try. Clearly, X has a geometric distribution with parameter Γ (according to our second assumption, Γ is constant in each try), and thus we have:

$$P(X = i) = \Gamma(1-\Gamma)^{i-1} \qquad (3)$$

and finally, we can calculate the *attack effectiveness* for a single piece as follows:

$$\begin{aligned}\text{AE} &= \frac{\text{expected download time during attack}}{\text{expected download time with no attack}} = \frac{E[T\times X]}{E[T]}\\ &= E[X] = \frac{1}{\Gamma} \approx \left(\frac{N}{N-M\times Q}\right)^k,\end{aligned} \qquad (4)$$

where $E[X]$ is the expected value of random variable X, and T is the amount of time needed to download a single piece. Fig. 1 shows *attack effectiveness* versus number of attackers (M) for different swarm sizes. Clearly, it can be seen that the size of swarm has significant impact on the effectiveness of the attack. In larger torrent networks, more attackers are needed to have acceptable results. To help the reader, Table 1 represents the notations we used for developing the proposed model. In Section 4, we present the practical values of *attack effectiveness* in our large-scale active measurement study in different scenarios.

To complete the model, it is also necessary to determine the amount of bandwidth which is required by the attack. Again, we build the model based on a specific sample piece so that it can be generalized for any arbitrary torrent file. Based on the assumption from the previous case, on average, a normal peer sends piece requests to k unique neighboring peers. Out of these k piece requests, the probability of sending exactly i requests to attackers, has a Binomial distribution with parameters k and ρ:

TABLE 1
Symbol Notation and Description

Symbol	Description
n	number of victim's neighbors that claim to have a specific piece.
m	number of attackers within the victim's neighbors that claim to have a specific piece.
k	average number of unique peers that an ordinary client will connect to, for downloading one piece.
N	number of active peers in the swarm.
M	number of attackers within the swarm.
Q	ratio of number of neighbors an attacker has, to the average number of neighbors an ordinary peer commonly has.
T	amount of time needed to download a single piece.
B	block size.
Φ	a random variable indicating the number of piece requests sent to the attacker peers.
Ψ	expected amount of bandwidth required to pollute a piece of an arbitrary peer.

$$\begin{aligned} P(\Phi = i) &= P(\text{sending } i \text{ piece requests to attacker peers}) \\ &= \binom{k}{i} \rho^{i} (1-\rho)^{k-i}, \end{aligned} \tag{5}$$

where $\rho = \frac{M \times Q}{N}$ is the probability for a neighboring peer to be one of the attacker peers. As mentioned earlier in this section, a single fake block is enough to deteriorate a complete piece. Hence, it suffices to respond with a single fake block to each piece request. Let B denotes the block size specified in the torrent metadata file and Ψ the expected amount of bandwidth required to pollute a piece of an arbitrary peer. Consequently we have:

$$\begin{aligned} \Psi = E[B \times \Phi] &= B \times E\left[\sum_{i=0}^{k} i \binom{k}{i} \rho^{i} (1-\rho)^{k-i}\right] \\ &= Bk\rho. \end{aligned} \tag{6}$$

Note that, the total amount of bandwidth required for infecting a specific piece in the entire torrent network can be determined by multiplying Ψ to the number of innocent peers:

$$\begin{aligned} \text{Total Required Bandwidth} &= \Psi \times (N - M) \\ &= Bk\frac{M \times Q}{N}(N - M). \end{aligned} \tag{7}$$

Remarkably, it can be observed that by assuming M as a variable, based on Eq. (4), AE is proportional to $(\frac{N}{N-MQ})^{k}$, but based on Eq. (7), total amount of required bandwidth is proportional to $NM - M^2$, where $M < N$ in both equations. Clearly, the value of AE is increasingly greater than the required bandwidth for all M. It means that, anti-P2P companies can continuously gain more success if they hire more attackers, and there is no optimum point for the amount of resources used for the attack. Fig. 2 shows how required bandwidth changes according to the number of attackers, based on the proposed model. As intuitively expected, it can be seen that larger torrent networks require more bandwidth.

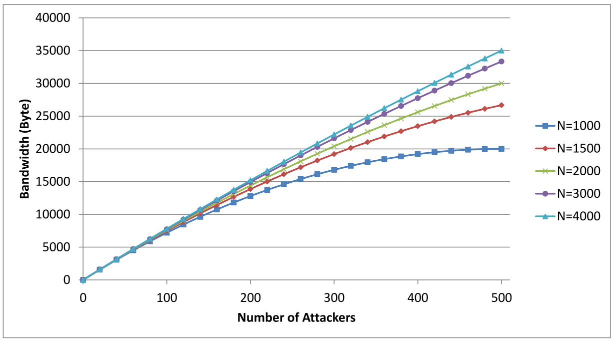

Fig. 2. Amount of bandwidth required to pollute an specific piece within a torrent network with different swarm size (N).

4 Large-Scale Active Measurement Study

In this section, we explain our measurement scenarios by which we attempted to evaluate the impact of Piece-Attack on different BitTorrent networks. In each scenario, we have repeated the experiment with different number of attackers with public IP addresses. By comparing the results obtained from each experiment and scenario, it is possible to determine the influence of different parameters (e.g., attack start time, number of attackers, number of public IP addresses used by attackers) on the effectiveness of the attack.

4.1 Attack Architecture and Preparation

The main goal of our measurement study is to observe the result of Piece-Attack in a completely real environment without any impact from artificial elements. To this end, we used uTorrent—as the most popular and common BitTorrent client in the world [20]—in our experiments. Since uTorrent is not an open source program, it is not straightforward to exploit it as an attacker peer to upload fake blocks in BitTorrent networks. To overcome this problem, we detected a simple, but effective gap inside the uTorrent application. To force uTorrent application to upload fake blocks, first we downloaded 70 percent of a torrent file with a uTorrent client and then we poisoned all the blocks within the downloaded files without any change in their sizes. This approach easily necessitates the uTorrent clients to act as an attacker, since they only check the correctness of a piece, right after it is downloaded, but they do not verify the healthiness of the pieces when they are going to upload data blocks to other neighboring peers.

To keep this phase simple but still effective, we considered the case in which the attackers contribute in uploading fake blocks throughout the attack period. However, one can consider the case where attackers pretend to be innocent ordinary users by uploading some flawless data blocks. Intuitively, it can help them to achieve a higher number of neighboring peers within the swarm. But as mentioned above, uTorrent is not an open source program and it is not straightforward to change its functionality.

4.2 Semi-Nat Protocol

To avoid allocating a whole host for every single attacker, we designed and developed an innovative *Semi-NAT* protocol to map a public IP address to every TCP/UDP connection coming out from our attackers. This protocol provides the ability of having numerous number of uTorrent clients (as attackers) in a single host, each of which have its own public IP address. The protocol implementation was totally separate from the host operating system (OS) which means that it was transparent to the uTorrent clients running on the attacker host. This worthy feature was necessary to provide an absolute real environment for the attack.

To make the protocol transparent, we must first manipulate the network routing of the host OS in order to route all the network traffic toward a fake network adapter (FN). This makes the host OS to think that it is connected to the Internet thorough the FN. Using jpcap library [21], the semi-NAT service captures the IP packets from the fake interface and translates each *new* TCP/UDP connection to an IP address. Available IP addresses can be specified manually or by using the DHCP multiple of times. After changing the source IP address of the captured packet, semi-NAT will send it through the origin network adapter (ON) which is already connected to the Internet gateway. In addition, it keeps a table to store previous mapped connections in order to use the same source IP addresses for all the packets from a single flow. By receiving a new packet on ON, semi-NAT simply replaces its destination IP address with FN's IP address. The modified received packet is then written on FN, so that the host OS thinks it received a new packet from the outside world which is connected to FN.

Practically, the whole process is executed completely hidden from the host OS's point of view, as it contemplates that FN is connected to the outside world. As an overview, the entire process is similar to the NAT protocol, since each transport layer connection is translated into a public IP address. The difference is that, there is no need to change the port numbers because the transport layer flows originally have unique port number, since a single OS is initiating them.

4.3 Measurement Scenarios

During our measurements, we repeated the experiments in various scenarios with different number of public IP addresses and attackers. In different scenarios we have aimed to figure out if it is possible to considerably influence the efficiency of downloading in a torrent network, with specific amount of resources allocated by our attackers.

We have launched the Piece-Attack over various torrent swarms to find out how powerful it is in hampering the distribution of copyright protected contents in different BitTorrent networks. For each scenario, we selected the proper torrent network in order to run multiple experiments on it. We ran one experiment at a time by starting each experiment immediately after the previous one, in order to keep the experiment parameters almost identical, during the whole scenario (e.g., number of seeds and leechers).

Although, there is no predefined protocol or a central entity in the BitTorrent protocol to keep track of malicious IP addresses, we have used two heuristic techniques to avoid our attackers to be blacklisted by the innocent users, in the targeted swarms: a) We have assigned random IP addresses to each experiment, b) In scenarios 2, 4, and 5, we have fired the experiments from lowest to highest based on the number of IP addresses used by the attackers. This technique guarantees to have fresh IP addresses in each experiment. Table 2 shows these scenarios.

TABLE 2
Measurement Scenarios for Different Torrent Network

Scenarios	Goal	Age of the Torrent
Scenario 1	measuring the effect of the number of attackers	Early Hours
Scenario 2	measuring the effect of the number of public IP addresses	
Scenario 3	measuring the effect of the number of attackers	Second Day
Scenario 4	measuring the effect of the number of public IP addresses	
Scenario 5	measuring the effect of the number of public IP addresses	Second Month

To monitor the impact of the attack on an ordinary victim peer, we used a single uTorrent client that downloads the targeted torrent file during the attack period. To make our observations short but still accurate, we just used our victim client for a constant time from 15 to 30 minutes for different measurement scenarios. During this time period, our victim can download about 50 pieces, which suffices to avoid any outlier results and also removes the necessity for multiple victim clients. Moreover, to avoid any unbiased results, we used a separate network in a different autonomous system (AS) for the victim client. Furthermore, to identify the effects of Piece-Attack on the BitTorrent network as a whole, we monitored the upload rate of our attackers during the experiments which will inform us about the amount of fake content that has been injected into the BitTorrent network. Next section will discuss the interesting results we have observed in different scenarios. Almost all the *Victim Results* are extracted from our victim's uTorrent GUI.

4.4 Measurement Results

In this section we present the results obtained during our active measurement study. We repeated our experiments over five different torrent files that have been chosen based on their ages (the time passed since their creation). For each experiment, we obtained the total number of leechers and seeds, based on our victim's GUI, to estimate number of nodes in the network.

Table 3 shows the results obtained during scenario 1. In this scenario we observed the impact of Piece-Attack on a torrent network in the first hours of its life with different number of attackers starting from 700 and ending to 1 with fixed number of public IP addresses used by them. In this scenario, our victim's 15 min download was totally within the attack period.

It can be seen that with only 200 active attackers, more than 90 percent of the victim's downloaded pieces are fake. Moreover, the *attack effectiveness* is above 10 which means that, during the attack period, the download time will prolong at least 10 times more than the download time without being under attack. Given that, it is obvious that the Piece-Attack is extremely powerful in hindering the distribution of torrent files if anti-P2P companies launch it in the first hours of torrent's existence. Furthermore, it is worth noting that, although the number of attackers doubled from the second row to the first row, we cannot see any significant

TABLE 3
Scenario 1: Measurement Results for Attacking a Torrent in the Early Hours of Its Life during a 15 Min Period

No.	Attack Resources		Victim Results						
	Public IPs	Attackers	Hashfails	Clean Pieces Downloaded	Seeders in Swarm	Leechers in Swarm	Wasted Time (min)	Fake Piece Ratio	Attack Effectiveness
1	250	700	36	1	29	112	14.3	97%	37
2	250	400	88	2	5	406	14.6	98%	45
3	250	200	99	10	27	603	13.6	91%	10.9
4	250	100	63	55	33	302	8	53%	2.15
5	250	50	97	15	40	160	13	87%	7.47
6	250	25	100	33	78	20	11.3	75%	4.03
7	250	12	102	53	63	39	9.9	66%	2.92
8	250	5	111	55	61	25	10	67%	1.02
9	250	1	61	105	40	10	5.5	37%	1.58

achievement in the results. This is because of the straightforward defense mechanism used not only by our victim uTorrent clients during the measurement study, but also by almost every popular BitTorrent client such as Vuze (Azureus). This defense mechanism blocks connections from two different peers with identical IP address. It means that, here, the victim has established connections to at most 250 of our attackers, even when we have many more attackers. Therefore, by increasing the number of attackers from 400 to 700 when we have only 250 public IP addresses, we should not expect extra achievement over a single victim peer. Although launching many more attackers than 250 will not have any influence on a single victim, it results in higher number of victim peers inside the network which results in more success if we see the whole BitTorrent network as a single victim.

In Fig. 3, a comparison between our model from Section 3 and the real results obtained from our measurement study, is presented. Undoubtedly, it shows how accurately the model can predict the *attack effectiveness* for a torrent network within the early hours of its life.

Table 4 shows the results for the second scenario. Again, the results are from launching the Piece-Attack against another torrent swarm in the first hour of its life. We repeated the experiment with different number of public IP addresses used by the attackers. The number of attackers is 100 for all the experiments. Similar to the prior scenario, it can be clearly seen that, the Piece-Attack can be very powerful if anti-P2P companies launch it against a torrent swarm as soon as it is created. The *attack effectiveness* in the first three rows shows that with 100 attackers who use at least a class C public IP address block, it is possible to prolong the download duration for an arbitrary torrent file, 10 times more than the duration in an ordinary situation. These results also confirm the fact that it is possible to have noticeable destructive effect on download time of clients with modest amount of resources used by the attackers.

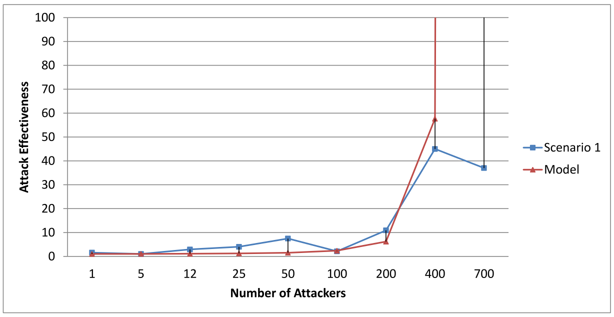

Fig. 3. Comparison between our model and data obtained during scenario 1. We used $k = 10, Q = 0.5$ and $N = 600$.

Tables 5 and 6 show the results of the third and fourth scenarios respectively. These results help us to understand how successful Piece-Attack is in hampering the illegal distribution of copyright protected materials when we launch it against torrent swarms that already have passed their first day of life. The key point about a torrent swarm when it passes a few tens of hours since it was born, is the ratio of seeds to leechers which become significant in proportion to the early hours of a torrent's life. This considerable number of seeds in the network will reduce the efficiency of our attackers. This is because our victim client, like other innocent peers in the swarm, prefers to download from seeds for avoiding any probable piece unavailability in the neighboring peers during the download time. In other words, in any BitTorrent network, it is vital to be sure that for every specific piece of the torrent file, you have more than a few neighboring peers who already have that piece. Piece unavailability occurs when we do not know a peer in the network who has a specific piece or those that already have that piece, leave the network suddenly. As a rule of thumb, seeds are the best choices for the goal of avoiding piece unavailability and that is because uTorrent clients try to find as much as possible seeds to select them as neighbors. Given that, the numerous number of seeds decrease the probability of our attackers to be selected as a source for downloading pieces by victim peers inside the torrent network. Therefore, it is reasonable that we do not have any measurement with *attack effectiveness* greater than 10 in contrast to the first two scenarios.

Interestingly, it can be observed that different amounts of resources used by the attackers did not have noteworthy effects on both *attack effectiveness* and *Fake Piece Ratio*, and their values are almost identical during different experiments in both scenarios 3 and 4. This fact can also be seen in the Fig. 4 where there is no relation between the *attack effectiveness* and number of attackers within the swarm.

It means that, even if anti-P2P companies use much higher amount of resources for launching Piece-Attack, they may not achieve any considerable success when the torrent has

TABLE 4
Scenario 2: Measurement Results for Attacking a Torrent in the Early Hours of Its Life during an 18 Min Period

No.	Attack Resources		Victim Results						
	Public IPs	Attackers	Hashfails	Clean Pieces Downloaded	Seeders in Swarm	Leechers in Swarm	Wasted Time (min)	Fake Piece Ratio	Attack Effectiveness
1	500	100	28	1	5	98	17.4	97%	29
2	250	100	34	4	5	106	16.1	89%	9.5
3	128	100	37	1	7	213	17.5	97%	38
4	65	100	22	12	5	114	11.6	65%	2.83
5	30	100	28	12	4	193	12.6	70%	3.33
6	15	100	20	36	5	115	6.4	36%	1.56
7	8	100	43	25	9	245	11.4	63%	2.72

TABLE 5
Scenario 3: Measurement Results for Attacking a Torrent in the Second Day of Its Life during a 15 Min Period

No.	Attack Resources		Victim Results						
	Public IPs	Attackers	Hashfails	Clean Pieces Downloaded	Seeders in Swarm	Leechers in Swarm	Wasted Time (min)	Fake Piece Ratio	Attack Effectiveness
1	100	500	40	49	150	21	6.74	45%	2.23
2	100	211	54	34	165	419	9.20	61%	1.63
3	100	100	47	49	144	613	7.34	49%	2.04
4	100	50	40	34	153	664	8.11	54%	1.85
5	100	25	54	40	157	43	8.62	57%	1.74
6	100	15	37	58	167	43	5.84	39%	2.57
7	100	6	31	54	172	49	5.47	36%	2.74
7	100	3	37	60	167	29	9.54	38%	2.62

TABLE 6
Scenario 4: Measurement Results for Attacking a Torrent in the Second Day of its Life during a 15 Min Period

No.	Attack Resources		Victim Results						
	Public IPs	Attackers	Hashfails	Clean Pieces Downloaded	Seeders in Swarm	Leechers in Swarm	Wasted Time (min)	Fake Piece Ratio	Attack Effectiveness
1	500	100	34	43	115	206	6.62	44%	2.26
2	250	100	44	73	116	205	5.64	38%	2.66
3	100	100	52	51	115	298	7.57	50%	1.98
4	50	100	20	95	115	219	2.61	17%	5.75
5	20	100	15	103	112	274	1.91	13%	7.87
6	10	100	15	75	119	306	2.5	17%	6.0

passed its first day of life. By a straight comparison between the results obtained during scenarios 3 and 4 against that of the first two scenarios, it is obvious that this is vital for anti-P2P companies to launch Piece-Attack against targeted torrent networks within the first hours since their creation. Otherwise, it is almost impossible to remarkably hamper the distribution of files in BitTorrent networks.

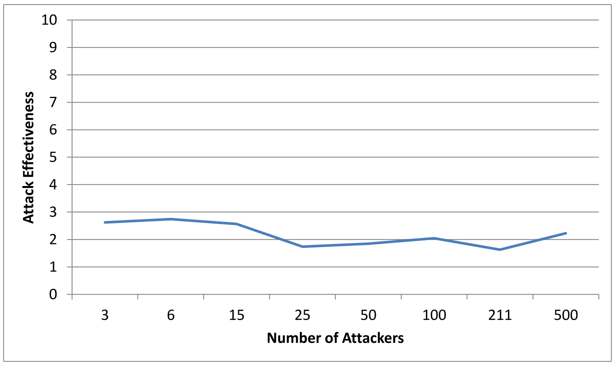

Fig. 4. *Attack effectiveness* versus M (number of attackers) in scenario 3 in which it passed one day since the targeted torrent creation.

As a proof of concept, Table 7 shows the results observed during the fifth scenario, where the attack has been fired against a torrent network, a month after the network was created. In this table, we see negligible random amount of hashfails for different number of public IP addresses used by 256 attackers.

In summary, we have launched Piece-Attack against three kinds of torrent networks. First, we attacked torrent swarms that were in the early hours of their life. Second, we repeated our measurements on the swarms that were in their second day and finally, on the swarms that passed their first month. We found that, it is not the amount of resources and bandwidth reserved for the attack, which is the most important factor in hassling the powerfulness of BitTorrent networks, but it is the time at which we launch the attack is what matters. Fig. 5 clearly proves that Piece-

TABLE 7
Scenario 5: Measurement Results for Attacking a Torrent in the Second Month of Its Life during a 30 Min Period

No.	Attack Resources		Victim Results			
	Valid IP addresses	Attackers	Hashfails	Clean Pieces Downloaded	Seeders in Swarm	Leechers in Swarm
1	128	256	0	43	6,128	8,088
2	64	256	0	73	5,970	12,800
3	32	256	3	51	9,910	1,175
4	16	256	1	95	9,193	1,166
5	9	256	5	103	12,580	2,385
6	2	256	2	75	12,534	2,334
6	1	256	2	75	12,481	2,269

Attack is significantly successful if it be used not after the day in which target torrent is created. We use *Golden Period* to point to the first hours of BitTorrent networks, since anti-P2P can exploit it in order to significantly affect the download time of end-users in BitTorrent networks with a few amount of resources. The observation from scenario 3, 4 and 5 verify our aforementioned claims.

It worth noting that torrent networks are mainly used for distribution of recently released top box office movies. Consequently, they attract a large portion of their users within the early hours of their creation, in which the demand for downloading is very high. Hence, Piece-Attack can play a considerable role in hampering the illegal distribution of file in those networks.

5 Conclusion

In this paper, we measured the impact of Piece-Attack on real BitTorrent networks. By launching large-scale Piece-Attacks against multiple real BitTorrent networks, we observed the success of the attack in prolonging the download time of end-users contributing to file sharing in the targeted networks.

According to the results, we discovered that anti-P2P companies can easily make the BitTorrent end-users to linger more than 10 times for downloading torrent files completely, only if they launch the Piece-Attack not after the *Golden Period* since the creation of the targeted swarm. We observed that even huge amount of resources used by those companies cannot hamper the capability of BitTorrent protocol in public distribution of copyright protected contents and BitTorrent networks are completely resilient against Piece-Attack if they have passed their first month.

As a future work, we intend to measure the impact of the Piece-Attack on BitTorrent networks for long-term periods to figure out the possibility of reducing the adding ratio of seeds in torrent swarms. Moreover, we expect different networks to react differently against the attack. Especially non-quantitative parameters such as popularity or IMDB rating are good candidates to analyze how various target movies resist against the Piece-Attack.

Acknowledgments

The authors would like to thank the *Center for Information Technology and Cyberspace, University of Tehran* for their support in providing required resources used in this work. Hamid R. Rabiee is the corresponding author.

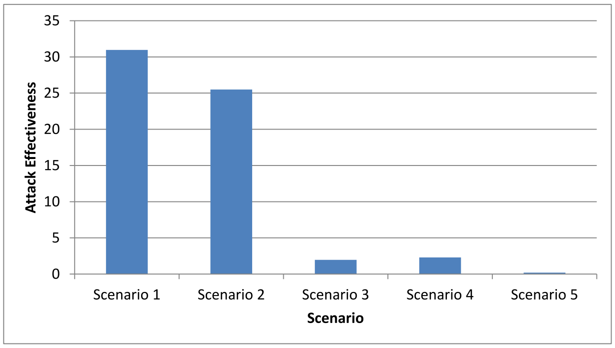

Fig. 5. Comparison between the average of *attack effectiveness* of first 3 experiments in each scenario. The results from first two scenarios are very impressive comparing with that of others.

References

[1] (2014, Mar. 16). An estimate of infringing use of the internet [Online]. Available: http://documents.envisional.com/docs/Envisional-Internet_Usage-Jan2011.pdf

[2] (2014, Mar. 16). Google trends [Online]. Available: http://www.google.com/trends/explore#q=bittorrent,kazaa,gnutella,edonkey,opennap&date=today%2012-m&cmpt=q

[3] (2014, Jan. 5). Bittorrent [Online]. Available: http://en.wikipedia.org/wiki/BitTorrent

[4] (2014, Mar. 16). The eclipse foundation open source community website [Online]. Available: http://eclipse.org

[5] (2014, Mar. 16). Linux.org website [Online]. Available: http://linux.org

[6] (2014, Mar. 16). Comparison of bittorrent clients [Online]. Available: http://en.wikipedia.org/wiki/Comparison_of_BitTorrent_clients

[7] P. Dhungel, D. Wu, and K. W. Ross, "Measurement and mitigation of bittorrent leecher attacks," *Comput. Commun.*, vol. 32, no. 17, pp. 1852–1861, 2009.

[8] G. Siganos, J. M. Pujol, and P. Rodriguez, "Monitoring the bittorrent monitors: A bird's eye view," in *Proc. 10th Int. Conf. Passive Active Netw. Meas.*, 2009, pp. 175–184.

[9] (2014, Mar. 16). Mediadefender's official website [Online]. Available: http://www.mediadefender.com

[10] (2014, Mar. 16). Copyright at MIT [Online]. Available: http://web.mit.edu/copyright/dmca-notices.html

[11] (2014, Mar. 16). File-sharing and copyright law [Online]. Available: http://acomp.stanford.edu/info/dmca/

[12] (2014, Mar. 16). Digital millennium copy right act (DMCA) [Online]. Available: http://www.dos.uci.edu/conduct/students/student-dmca.php

[13] S. Altschuller and R. Benbunan-Fich, "Is music downloading the new prohibition? what students reveal through an ethical dilemma," *Ethics Inf. Technol.*, vol. 11, no. 1, pp. 49–56, 2009.

[14] M. Piatek, T. Kohno, and A. Krishnamurthy, "Challenges and directions for monitoring P2P file sharing networks, or, why my printer received a DMCA takedown notice," Dept. Comput. Sci. Eng., Univ. Washington, Seattle, WA, USA, Tech. Rep. 08-6-01, 2008.

[15] (2014, Nov. 13). Semi-nat project [Online]. Available: https://github.com/fattaholmanan/semi-nat

[16] M. A. Konrath, M. P. Barcellos, and R. B. Mansilha, "Attacking a swarm with a band of liars: Evaluating the impact of attacks on bittorrent," in *Proc. 7th IEEE Int. Conf. Peer-to-Peer Comput.*, 2007, pp. 37–44.

[17] W. Haizhou, C. Xingshu, and W. Wenxian, "A measurement study of polluting a large-scale p2p IPTV system," *China Commun.*, vol. 8, no. 2, pp. 95–102, 2011.

[18] J. SHI and H. ZHANG, "A protocol based countermeasure to bittorrent fake-block attack?" *J. Comput. Inf. Syst.*, vol. 8, no. 12, pp. 5211–5218, 2012.

[19] (2014, Mar. 16). Bittorrentspecification [Online]. Available: http://wiki.theory.org/BitTorrentSpecification

[20] (2014, Mar. 16). Google trends [Online]. Available: http://www.google.com/trends/explore#q=utorrent,azureus,flashget,Bittorrent,BitTyrant

[21] (2014, Nov. 13). Network packet capture library for applications written in java [Online]. Available: https://github.com/jpcap/jpcap

Ali Fattaholmanan received his BSc degree in software engineering from Sharif University of Technology, Tehran, Iran, in 2011. He is currently conducting research on computer networks and network security as an MSc student at Digital Media Laboratory, Sharif University of Technology. His research interests include computer networks, peer-to-peer networks, and network security.

Hamid R. Rabiee (SM"07) received his BS and MS degrees (with great distinction) in electrical engineering from CSULB, the EEE degree in electrical and computer engineering from USC and the PhD degree in electrical and computer engineering from Purdue University, West Lafayette, in 1996. From 1993 to 1996, he was a member of Technical Staff at AT&T Bell Laboratories. From 1996 to 1999, he worked as a senior software engineer at Intel Corporation. He was also with PSU, OGI, and OSU universities as an adjunct professor of electrical and computer engineering from 1996 to 2000. Since September 2000, he has joined Sharif University of Technology (SUT), Tehran, Iran. He is the founder of Sharif University Advanced Information and Communication Technology Research Center (AICT), Advanced Technologies Incubator (SATI), Digital Media Laboratory (DML), and Mobile Value Added Services (MVAS) laboratories. He is currently a professor of computer engineering at Sharif University of Technology, and the director of AICT, DML, and MVAS. He has been the initiator and director of national and international level projects in the context of UNDP International Open Source Network (IOSN) and Iran's National ICT Development Plan. He has received numerous awards and honors for his Industrial, scientific and academic contributions, and has acted as chairman in a number of national and international conferences, and holds three patents. He is a senior member of IEEE.