Interception BitTorrent Traffic in A University Network

Merouane MEHDI

Electronics Department, University Blida1, Algeria.
Route Soumaa BP 270 BLIDA

Abstract
Nowaday, many universities are faced with bandwidth saturation problem caused by several parameters namely using youtube abusively, online games and especially illegal downloading makes havoc with the use of Peer-to-Peer protocol such as BitTorrent, which is often associated with data piracy and copyright violation. This article aims to present the one hand the impact of the use of Peer-to-Peer file sharing traffic on campus bandwidth by observation of BitTorrent traffic and the other hand to provide a method for limiting the illicit access at this kind of network. For this work, we preferred to use a set of open source tools like wireshak sniffer to capture Bitorrent traffic, and using the famous Snort intrusion detection with the creation of new rules to detect and alert this traffic on campus and take steps to bypass this. The solution allowed us to have a bandwidth saturation reduction of 35%.

Keywords
Intranet, P2P traffic, BitTorrent, µtorrent, Bandwidth, Snort IDS.

1. Introduction

The development of computers and the computer network coupled to the democratization of Internet access facilitate the dissemination of information in a digital form elusive. Each individual connected to the Internet can access a wealth of varied information. Today, almost everything goes through Internet: our message, content that we consult on the Web but also television, live video, etc. This makes data gigabits; the Peer-to-Peer (P2P) takes a large part of this network.

Peer-to-peer (P2P) is a decentralized communications model in which each party has the same capabilities and either party can initiate a communication session. Unlike the client/server model, in which the client makes a service request and the server fulfills the request, the P2P network model allows each node to function as both a client and server [2].

Is it estimated that for any given network 60 to 80% of their traffic is consumed by P2P traffic. So even in your office, if people are using P2P application they will consume a huge amount of bandwidth. P2P application is very famous for distributing Pirated software. Your users might be using pirated software on their computers and Auditors will never appreciate that. You can never trust the file you are downloading from a remote user in P2P environment and 90% of the files contain malwares. Thus if your users are using P2P application there is very high rate of Virus Outbreak in your network that too very frequently. In 2008, 10% of malware were propagated via P2P applications. Even the very infamous W32.Downadup also propagated and updated itself via P2P applications [2].

BitTorrent is a communications protocol of peer-to-peer file sharing ("P2P") which is used to distribute data and electronic files over the Internet. BitTorrent is one of the most common protocols for transferring large files, such as digital video files containing TV shows or video clips or digital audio files containing songs. Peer-to-peer networks have been estimated to collectively account for approximately 43% to 70% of all Internet traffic. In November 2004, BitTorrent was responsible for 25% of all Internet traffic. As of February 2013, BitTorrent was responsible for 35% of all worldwide bandwidth [3]. In early 2015, AT&T estimates that BitTorrent represents was responsible for a quarter of all Internet traffic. BitTorrent and µTorrent software client surpass 150-million user milestone [4].

The impact of using µTorrent client made more havoc on the bandwidth that an attack of denial of service (DoS), since this is a p2p downloading not detected by the IDS and function full time, although all the expense of files size has downloaded and the number of users. The size of files to download from P2P network continues to growth, such as video, particularly the movie download that switches a 720x300 resolution for a 700 MB to a current size of 20GB or more for a 4K resolution, same thing for games. As of today, the size can go beyond 60 GB for games, imagine the occupation of bandwidth that can be taken to download this kind of file. A central server in a campus could not easily bear

Manuscript received June 5, 2026
Manuscript revised June 20, 2026
https://doi.org/10.22937/IJCSNS.2026.26.6.11

such amount of information, also the growing number of sites offering downloads P2P continues to grow; currently: sites like RARBG.to or t411.ch and many other sites that offer any kind of illegal downloading with p2p network. For now, I speak only of p2p downloading but this network is working in both directions for uploading and downloading, the problem is more complicated.

It is in this context that comes our study, in our university and specifically in the Academic Research Network (ARN), we recorded a net debit regression, HTTP and FTP downloads are virtually stopped due to p2p. Different institutions use a bandwidth of 100 MB, including ours. The problem forces us to look closely, firstly, the impact of p2p networks on bandwidth, and secondly to find a real solution for this problem.

At this effect, we have taken as detection tool the most known Snort intrusion detection software in the aim of creating adequate rules to bypass this uTorrent download, reduce the consumption of bandwidth within our university and subsequently generalized to the entire ARN network. The challenge, however, is in detecting the P2P file sharing programs, tracker site and stopped them. Knowing at first, the incoming and outgoing traffic for peer to peer file sharing occupy up to 10% (upload) and 34% (download) bandwidth in our university campus (Figure 1) during working hours, the number of users between teachers, workers and students exceed 50,000 with a number of 3000 simultaneously connected.

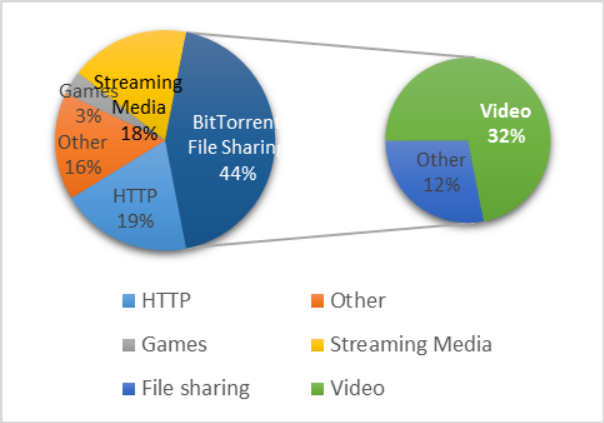

Figure.1 Bandwidth Using in the Campus

In order to achieve this main objective, several steps were pursued:

1. Describe the mechanism used by BitTorrent and uTorrent client.
2. Sniff the traffic using sniffer tool to capture BitTorrent packets.
3. See the impact on bandwidth campus
4. Attempts to create snort rules to detect traffic P2P.
5. See the challenge of data encrypted.
6. Identify the BitTorrent protocol and particularly the applications.

‘’ The ARN network was deployed in the early 90s to provide a technological infrastructure for the benefit of all stakeholders in higher education, scientific research and technological development.’’

2. BitTorrent Protocol

Programmer Bram Cohen, a former University at Buffalo student, designed the protocol in April 2001 and released the first available version on 2 July 2001 and another version in 2013. BitTorrent clients are available for a variety of computing platforms and operating systems including an official client released by BitTorrent, Inc [5].

The BitTorrent network has a very particular architecture. It is not centralized as was Napster or eDonkey2000, connecting all the peers on one big server. The main risk is that if the server falls involves the entire network. It can also entered by the court if it considers that this server contains files subject to copyright. That is what happened to Napster, which was closed down in 2002 [5].

It does not look like a decentralized architecture, which it is connecting peers on multiple servers simultaneously (Emule, Fast Track ...). These servers communicate with each other; the system will be more robust, but still.

In fact, the peers are not part of a global network, but they are grouped by file. There is a network around each “.torrent” file. BitTorrent is a set of mini-networks, and it is connected only to users with the data you want to download and / or share.

To send or receive files, a person uses a BitTorrent "client" on his Internet-connected computer. A BitTorrent client is a computer program that implements the BitTorrent protocol. Clients include µTorrent, Xunlei, Transmission, qBittorrent, Vuze, Deluge, and BitComet. BitTorrent trackers provide a list of files available for transfer, and allow the client to find peer users known as seeds who may transfer the files. In

our case, we take µTorrent client, 60% of users in the campus download with this client.

A. BitTorrent anatomy:

At BitTorrent architecture, there are two aspects [6]:

1) Website hosting called very often BitTorrent tracker website whose operation is as follows:

- Start running a tracker.
- Start running an ordinary web server, such as IIS, or have one already.
- Associate the extension .torrent with mimetype application/bittorrent on their web server.
- Generate a metainfo (.torrent) file using the complete file to be served and the URL of the tracker.
- Put the metainfo file on the web server.
- Link to the metainfo (.torrent) file from some other web page.
- Start a downloader, which already has the complete file (the 'origin').

2) The second aspect that interests us is the downloader, the user downloads p2p file following these operations:

- Install BitTorrent client.
- Surf the web. Enter in the BitTorrent website.
- Click on a link to a .torrent file.
- Select where to save the file locally, or select a partial download to resume.
- Wait for download to complete.
- Tell downloader to exit (it keeps uploading until this happens).

Before understanding the traces of BitTorrent and µTorrent in captured packets, we will give an overview of the different terminology related BitTorrent:

Index	An index is a list of .torrent files managed by a website and available for searches. An index website can also be a tracker.
Peer	A peer is one instance of a BitTorrent client running on a computer on the Internet to which other clients connect and transfer data.
Torrent	A torrent can mean either a .torrent metadata file or all files described by it, depending on context. The torrent file contains metadata about all the files it makes downloadable, including their names and sizes and checksums of all pieces in the torrent.
Seed	A seed refers to a machine possessing some part of the data. A peer or downloader becomes a seed when it starts uploading the already downloaded content for other peers to download from.
Leech	The term leech also refers to a peer (or peers) that has a negative effect on the swarm by having a very poor share ratio, downloading much more than they upload.
Swarm	Together, all peers sharing a torrent are called a swarm.
Tracker	A tracker is a server that keeps track of which seeds and peers are in the swarm. Clients report information to the tracker periodically and in exchange, receive information about other clients to which they can connect.

B. P2P file sharing program "µTorrent".

A µTorrent client is program that implements the BitTorrent protocol. Each client is capable of preparing, requesting, and transmitting any type of computer file over a network, using the protocol. A peer is any computer running an instance of a client. To share a file or group of files, a peer first creates a small file called a "torrent". This file contains metadata about the files to be shared and about the tracker, the computer that coordinates the file distribution [8]. Peers that want to download the file must first obtain a torrent file for it and connect to the specified tracker, which tells them from which other peers to download the pieces of the file (figure 2).

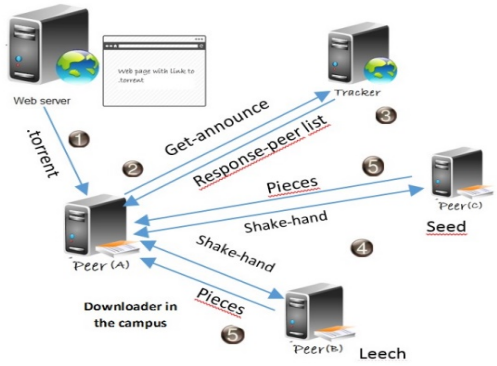

Figure.2 BitTorrent File Sharing Network.

3. Materials and Methods

We use network monitoring to collect traces of traffic flowing between the University and the rest of the Internet. Campus network connects to its ISPs via a border router; its use for outbound and inbound traffic. This router is connected to switch. This switch has a monitoring port that is used to send copies of the incoming and outgoing packets to our monitoring host. The entire DMZ passes through the switch, the users of the campus network automatically pass through the proxy server having a local address. The only means of security is the firewall that sits between the router and the switch.

At the monitoring host noting us the magnitude of the consumption of bandwidth each day for a period of one month with PRTG graph software, and for BitTorrent Activity on a campus Network, we use WireShark sniffer software (Figure 3).

By cons for understanding the phenomenon P2P, we were installed a BitTorrent client more accurately Utorrent on a PC with a public address connected to the switch. With the help of sniffer Wireshark, we will show in what follows various clues to detect this p2p download in the intranet of the campus.

A site like RARBG.to that is very popular includes several types of files to download freely namely the music, books, games, applications and especially high-quality movies. This choice to website is only to give and explain how this kind of torrent tracker works, especially not for free advertising.

Through this domain can simply make a ping to determine the IP address, the result gives the address 185.37.100.122 for domain “rarbg.to”. As can be easily observed at the capture. The connection to the website is through a link mentioned here https://rarbg.to, just click on one of the index mentioned in the website page the user starts downloading through torrent client. The file has downloaded has an extension ".torrent".

The test performed with the client µTorrent, the last release of software is 3.4.8. A BitTorrent client normally associates the TCP port number 6881. However, if this port is busy for some reason, the client will instead try successively higher ports (6882, 6883, and so on up to a limit of 6999). With the campus network firewall, we can already start by putting a rule to block these ports. In addition, we can observe the BitTorrent client (µTorrent) port as 60447 (Figure 5), which is being communicated to BitTorrent Server as HTTP Request.

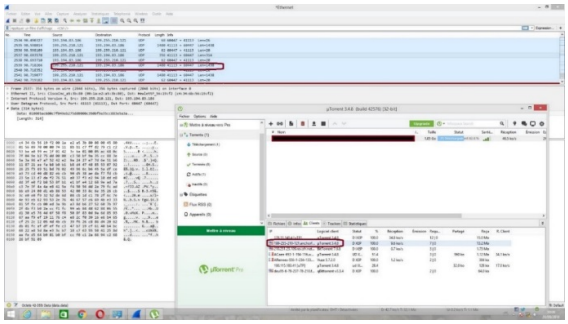

Figure.3 BitTorrent Traces in the Wireshark sniffer software

A. Tracker Websites P2P access

Start with detected the different torrent tracker website for the network administrator is a primary thing, A recent study dated September 2016, regroups the most popular tracker web p2p, see table 1.

Table 1. Popular tracker P2P websites

Demonoid.cc	**RARBG.to**	EZTV.ag	idope.se
Bitport.io	Boxopus	ExtraTorrent.com	Toorgle.com
SeedPeer.au	Isohunt.to	Torrent Funk	Torlock:

At our campus, wireshark allowed us just to see that the majority of torrent tracker websites are mentioned in this table 1, add to this table a torrent tracker website often consulted by campus users, not known to English speakers as it is a French website "cpasbien.cm". The screenshot below clearly shows the name of torrent tracker website in the sniffer traces (Figure 4).

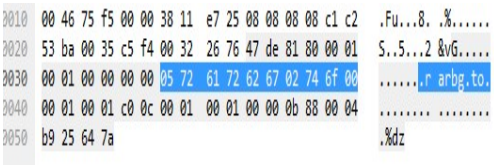

Figure.4 Trace of BitTorrent tracker site

Looking at sniffer traces (Figure 5), we noticed that the torrent “announce” requests over HTTP GET by a torrent tracker is visible clearly; see the screenshot in the figure 5. To generalize the information that interests us comes just before the string utorrent is "User-Agent", this string precedes every BitTorrent client for http get “announce” or “scrape”.

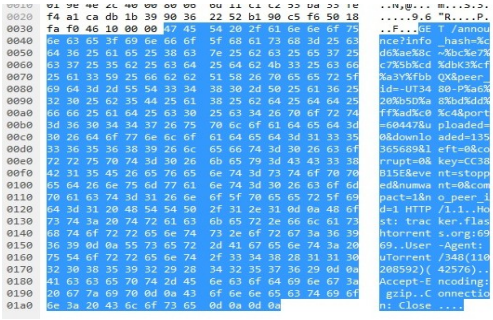

Figure.5 Torrent metafile content “announce”

B. Capture metafile “.torrent”

Tests were performed to understand the signatures of BitTorrent protocol and traces of Torrent client. We took the case of a download from a website Tracker by hazard the file « randonnées en France ». When downloading the configuration of µTorrent is by default, the figure 6 below shows clearly the name of the metafile with HTTP Get sniff with wireshark software. Encryption default mode is not enable.

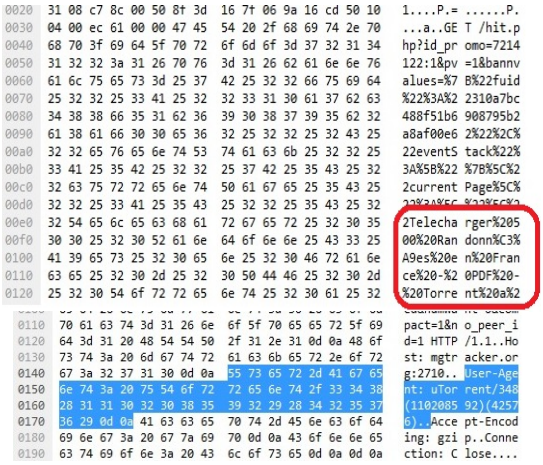

Figure.6 Torrent metafile capture

Furthermore, even the client BitTorrent "uTorrent" can be observed when sniffing packets with the version 3.4.8 build 42576. Once the user have a connection to peer(s), the first message he sends should be a Shake-hand. For the current protocol, ‘pstrlen’ = 19 and ‘pstr’ = ‘BitTorrent protocol’ [10]. This string is visible in the sniffer traces (figure 7).

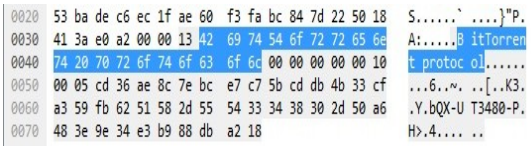

Figure.7 BitTorrent shake-hand capture

So far, the client uTorrent works with a Distributed Hash Table (DHT) disabled, knowing that the DHT is used by BitTorrent clients to find peers via the BitTorrent protocol. Once the DHT is activated, a ping is used to look after the peers [11]. So there can be detected in tests performed and following, DHT ping is represented by this string ‘’d1:ad2:id20‘’ follows with “ping” implemented over UDP protocol, the figure 8 clearly shows this appearance. The infohash of the torrent mentioned by “info_hash20” in the sniffer traces associated with a getpeers represented by “get_peers1”.

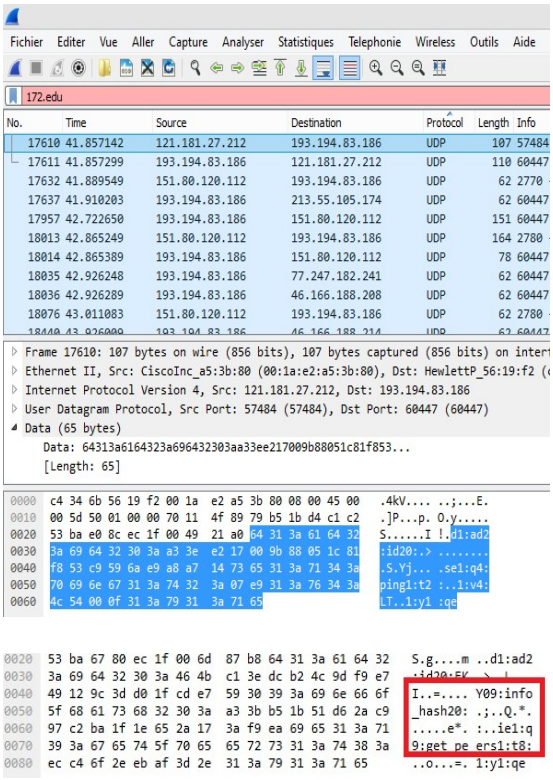

Figure.8 BitTorrent client with DHT enable

C. *File sharing encrypting*

In the detected signatures campus prove that the exchange of information in the µTorrent is unencrypted, but that does not stop to see the encryption aspect, advanced users have the option of encrypted information so that it is not detectable. Once encrypted, all peer-to-peer exchanges will be encrypted. P2P information exchanged will be only between BitTorrent clients support encryption, automatically many sources and peers are reduced and the user have a more difficult to download files, say it reduces the occupation of campus bandwidth.

So with the current version of wireshark version 2.0.5, we could detect even by encrypting the exchange, signatures utorrent in this form, see figure 9 below.

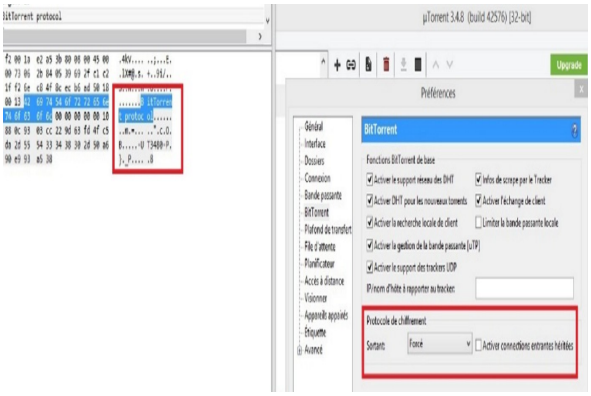

Figure. 9 Forced encrypting BitTorrent protocol

The image (figure 10) shows clearly the signature "BitTorrent Protocol" in captured packets, followed by the version of µTorrent used by the user in this form "UT3480-P". The traces of the ".torrent" metafile are undetectable because of encryption. With this information, the rules developed in Snort can easily be used.

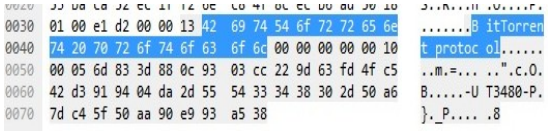

Figure.10 BitTorrent protocol capture

4. SNORT Rules

The Snort intrusion detection is free and runs on any modern operating system and any old hardware you have, Snort rules define the patterns and criteria it uses to look for potentially malicious traffic on our network. Without these IDS rules, Snort is just another sniffer like wireshark. Writing Snort rules is made in a simple language, follow this scheme:

The header of rule that contains:

1. The protocol used for data transmission.
2. The action of the rule.
3. IP source and destination addresses and mask.
4. The source and destination ports.

The rule options (brackets) that contain:

1. The alert;

2. The conditions determining the sending of the alert depending on the inspected package.

The next section explain how to create a customized rule for local use.

The Rule Options section:

Alert is the defined action when a matching signature is detected. The signature in this case is the presence of predefined flags set in the TCP header. Signatures within other rules may be matching payload content, other flags, or binary data (Figure 11).

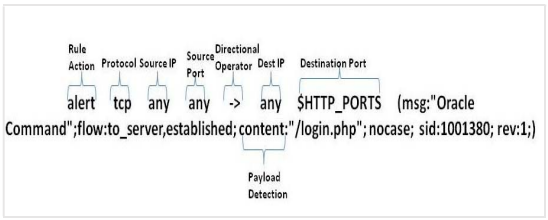

Figure. 11 Snort rule description

msg:

The message option explains the type of activity being logged. It is a way for the rule's author to better explain the reason for the alert.

content*:*

The content option is a keyword for defining stings of text or hexadecimal data within the payload. This option is case-sensitive, but can be used with the nocase modifier for case-insensitive matching.

Nocase:

The content modifier nocase deactivates case-sensitivity and looks for matching content.

sid:

This keyword is used to uniquely identify Snort rules.

Classtype: this keyword is used to categorize a rule as detecting an attack that is part of a more general type of attack class.

The following sample rule is simple and can detect login attempts as root for the telnet protocol (port 23):

```
alert tcp any any -> 192.168.1.1/24 23
(msg: " Telnet access attempt for root
";content: "USER root"; nocase;)
```

A. Detecting BitTorrent in the campus

In the previous section, the sniffer wireshark revealed several clues in the captured packets to the use of BitTorrent the following table 2 shows all indices. These indices will enable us to create specific rules in Snort to detect the BitTorrent content.

TABLE1. BITTORRENT CONTENT DETECTION

BitTorrent Message	**BitTorrent content "string"**
Torrent Tracker website	rarbg " all tracker web site here" GET /announce /scrape
BitTorrent shake-hand	BitTorrent protocol
Client BitTorrent	Torrent uTorrent User-Agent T3480-P d1:ad2:id20 ping info_hash20 get_peers1

All this information will allow us to develop new rules Snort in order to detect the BitTorrent content. The following rules were implemented:

Detecting Torrent website tracker "case: rarbg.to":

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg:
"Torrent tracker rarbg";
content:"GET";
content:"rarbg"; track
by src; sid:1000502; rev:2;)
```

This rule is applied to all the torrent tracker websites most popular cited above. The tracker BitTorrent website "cpasbien.cm" is one of the first sites to be detected here at the campus.

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg:
"Torrent tracker
cpasbien.cm"; content:"GET";
content:"cpasbien"; track
by src; sid:1000510; rev:2;)
```

Detecting metafile torrent and announce string in the HTTP GET incoming and outcoming

```
alert udp $HOME_NET any ->
$EXTERNAL_NET any (msg:”P2P
DHT enable”;
content:”d1:ad2:id20”;
content:”ping”;
classtype:policy-violation;
sid:1000506; rev:2;)
```

```
alert udp $HOME_NET any ->
$EXTERNAL_NET any (msg:”P2P
DHT get peers”;
content:”d1:ad2:id20”;
nocase;
content:”info_hash20”;
nocase; content:”get_peers1
classtype:policy-violation;
sid:1000516; rev:2;)
```

```
(msg:”BitTorrent http
request out”;
flow:to_server,established;
content:”GET”;
content:”/announce”;content:
”info_hash”;
classtype:policy-violation;
```

BitTorrent client Shake-hand incoming and outcoming

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any
(msg:”BitTorrent shakehand
in”;
flow:to_server,established;
content:”BitTorrent
Protocol”; classtype:policy-
violation; sid:1000505;
rev:1;)
```

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any
(msg:”BitTorrent shakehand
out”;
flow:from_client,established;
content:”BitTorrent
Protocol”; classtype:policy-
violation; sid:1000515;
rev:1;)
```

BitTorrent client with DHT enabled

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any
(msg:”BitTorrent metafile”;
flow:to_server,established;
content:”GET”;
content:”torrent”;
classtype:policy-violation;
sid:1000503; rev:2;)
```

Using the 3.4.8 version of uTorrent is answered in the campus detect this customer, is easy with this rule:

```
alert tcp $HOME_NET any ->
$EXTERNAL_NET any (msg:”P2P
Utorrent”;
flow:to_server,established;
content:”User-Agent:
uTorrent”; classtype:policy-
violation; sid:1000508;
rev:1;)
```

With different BitTorrent client, we change only the name of torrent application.

5. Discussion

The network administrator needs a monitoring system that will allow it to have a view details on network transactions within its network, the campus suffered for some years a huge network saturation, the study was done shown, the impact of P2P on bandwidth. The first step was understanding this BitTorrent appearance on the network, the sniffer wireshark allowed us to watch closely the signature of this P2P network constituted firstly the Bittorrent client software installed at the user and secondly tracker web site offering such metafile torrent download.

The user of the campus has a variety of customer software p2p "qbittorrent, vuze, µTorrent, Transmission, Tixati, Deluge and many other", the study focused on utorrent client seen using it improperly on campus, different rules have been developed for the detection of those signatures, it is generalized for other clients.

Detecting P2P tracker website is to detect sites that link to the download file ".torrent", the website "rarbg.to" is an example, but in reality full of other websites that offers the same, and they are still in full

growth, therefore the development of rules to detect this kind of website can also be generalized to other websites. At the campus we have developed about 50 rules regroups popular websites tracker and BitTorrent clients. The proportion of snort rules trigged for BitTorrent traffic in the campus for 3 Days in working hours (Figure 12). The number of alerts proves the real using of this file sharing P2P in the campus. The consultation of torrent website is always growing.

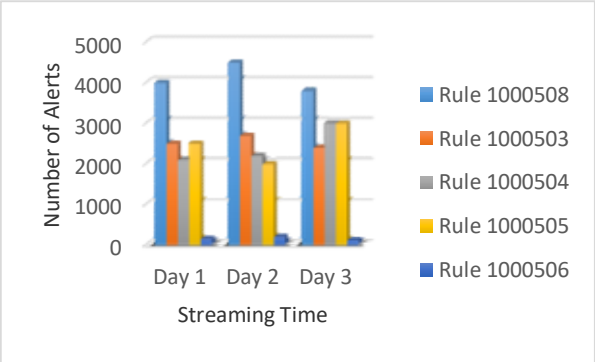

Figure.12 Proportion of Snort rules triggered for Bitorrent traffic in the campus

The proposed solution to block these alerts is every first set up a user management strategy of the campus, by setting up a user directory for access to the Internet in the proxy server. Implemented a rule deny for eliminate sites that link with the p2p file, for the BitTorrent client used warning to those who used them since each user with the directory using this kind of software is detected. Otherwise, the use of firewall to block ports and even unauthorized IP addresses.

With this strategy in place, we have seen a marked improvement in browsing the internet on campus, the proof is shown in figure13, which shows bandwidth before and after elaborations rules. Despite this effort, the impact of this work is minimal since the ARN network includes 134 institutions connected to the same backbone. The application of this modest work in all institutions is a powerful solution for eliminating this P2P download.The figure below shows the traffic, variation for a month before setting up Snort rules and after development of these. Thus we can see the real impact that bandwidth suffer daily. A clear regression bandwidth occupation of BitTorrent traffic, the figure 13 shows this.

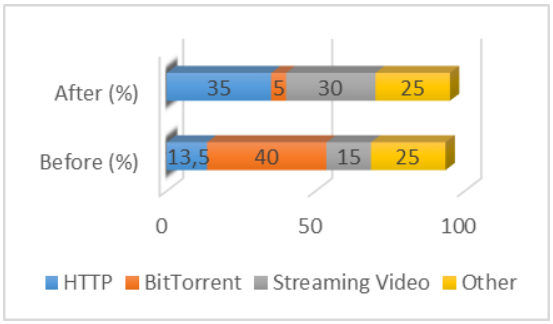

Figure.13 Campus bandwidth categories using

6. Conclusion

This document has a goal to show the impact of the use of P2P to the university campus such as BitTorrent and uTorrent client with turnkey solution for that. Auscultate traffic of the campus network has allowed us to observe the one hand the growing number of use of the BitTorrent client that is about 60% and secondly identify traces and anatomy of this protocol with the tool wireshark.

This knowledge has enabled us to develop a strategy based on the creation of new rules for P2P detection in the Snort IDS and the use of the campus firewall to block these sites and BitTorrent client software. The misuse of such a network automatically implies a saturation of bandwidth; torrenting consumes a huge amount of bandwidth. BitTorrent traffic is possible to detect accurately both TCP and UDP traffic but it is hard to be blocked completely, it's because each download/upload are not from a specific IP address but rather to an infinitely many IP addresses. It has also proven that even with encryption some clues to detect this type of use on campus. The security strategy implemented, allowed us to reduce the consumption of bandwidth by 35%. After a month of implementation of this strategy, we observed another phenomenon that has passed, as some network users have found a solution to these rules away, with the help of TOR network, precisely in the future work we will proceed to find how to bypass this problem.

Références

[1] Cook, T., Conti, G., Raymond, D. When good Ninjas turn bad: Preventing your students from becoming the threat. *Proceedings of the 16th*

Colloquium for Information System Security Education, 2012, 61-67.

[2] Friend Recommending Peer-To-Peer File Sharing and Synchronization Application, International *Journal of Scientific Research and Engineering Studies (IJSRES)*. Vol 2 Issue 3, March 2015.

[3] "Application Usage & Threat Report". *Palo Alto Networks. 2013.* Archived *from the original on 31 October 2013*. Retrieved 7 April 2013.

[4] "AT&T patents system to 'fast-lane' BitTorrent traffic". *Thestack.com.* 8 May 2006. Retrieved 5 March 2015.

[5] Pooja Balhara , A Review on Torrent & Torrent Poisoning over Internet. *International Journal of Computer Science & Management Studies*, Vol. 22, Issue 01, 2016.

[6] Ying-xu, Hong-guo Yang. Research on Client Detection of BitTorrent Based on Content. *Communications in Computer and Information Science. Springer* pp 473-476, vol 215, 2011.

[7] Ang, Liang; J., Kangasharju,. "Measuring large-scale distributed systems: Case of Bit Torrent Mainline DHT". *IEEE P2P 2013 Proceedings*. pp. 1–10.

[8] "BitTorrent and µTorrent Software Surpass 150 Million User Milestone". *Bittorrent.com*. 9 January 2012. Archived from the original on 26 March 2014.

[9] Eric Andre, "How to fight P2P in a corporate environment", SANS Institute 2004, GSEC Practical V1.4b, Case study in information security. Technical report.

[10] Richard wanner "Detecting Torrents Using Snort", SANS Institute 2009, Technical report.

[11] Andrew Loewenstern, Arvid Norberg. "DHT Protocol", BitTorrent.org. URL: http://www.bittorrent.org/beps/bep_0005.html, last access in March 22. 2013.

[12] Poo Kuan Hoong, Hiroshi Matsuo, A Super-Peer based P2P Live Media Streaming System, Proceedings of the World Congress on Engineering and Computer Science 2007 WCECS 2007, October 24-26, 2007, San Francisco, USA.

[13] Li Yang, Daiyun Weng, Snort-based Campus Network Security Intrusion Detection System, *Information Engineering and Applications, Springer*, 2012, pp. 202-208.