P2P Implications on Web Surfing

Merciadri Luca (Luca.Merciadri@student.ulg.ac.be)

Abstract

P2P (Peer-to-Peer) technology comprises various ways to exchange information rapidly, each participant sharing a portion of his own resources ([2]). However, despite of the numerous advantages of using P2P, a real problem is often encountered when using cheap internet connections: web surfing becomes so slow that it seems impossible to reach a web page, for the P2P's user. It is especially the case when using connections with a low upload speed. The problem has also an importance, even if it is minor, when using high-speed connections (VDSL, ...), as it is also a waste of capacity.

Introduction

P2P (Peer-to-Peer) technology comprises various ways to exchange information rapidly, each participant sharing a portion of his own resources ([2]). Anyway, despite of the numerous advantages of using P2P, a real problem is often encountered when using cheap internet connections: web surfing becomes so slow that it seems impossible to reach a web page, for the P2P's user. It is especially the case when using connections with a low upload speed. The problem has also an importance, even if it is minor, when using high-speed connections (VDSL, ...), as it is also a waste of capacity.

Causes

This phenomenon is principally due to two concepts intrinsically linked to the user's P2P client:

1. The P2P client maintains lots of connections with other P2P clients and servers, for various purposes,
2. The P2P client tries to upload at the *maximum* speed it can upload, so it "shares" as much as it can with P2P's clients. This *maximum* speed is either (manually, or automatically[a]) set in the P2P client's options, or is simply the *maximum* the ISP (Internet Service Provider) allows its user to use.

Unexperimented users do not modify the DownLoad (DL)/UpLoad (UL) rules, as they are often available (only) in special panels, except for BitTorrent clients. It is also disadviced for beginners to modify them, as they appear to be "critical" settings, the efficiency of their P2P client greatly depending upon it. Most of P2P clients-servers use as *maximum* speeds the *maximum* available bandwidth (essentially for upload, download speed being sometimes limited by default, to prevent users with connections having high bandwidths from downloading too much). Thus, every user who does not modify these settings uses a client which uploads and downloads at the highest speed it can, depending on the available bandwidth.

Implications

However, when a P2P's user wants to visit a website, he needs (or will resign to his fate) to stop these downloads, if his P2P client-server is working, as it takes too much bandwidth, and engender too much connections. Regrettably, stopping these downloads results in lost connections with other P2P's users. As there are very often queues for overwhelmed servers[b], the P2P's user who simply wanted to visit a website loses his progression in others' queues. He will thus download at a potentially lower speed when using his P2P client on its next restarting, assuming it is restarted just after he visited the page that he reached instantly.

That is a pretty waste of time, and of computer resources. It should be avoided. Although a P2P client has to establish lots of connections, P2P should *not* avoid web surfing. Think about future: it is normal (*i.e.* a desirable thing) to download legal files, such as legally-downloadable TV series, or free books, using P2P, when surfing, both simultaneously, isn't it?

The problem could appear simple, but it is not the case, as the P2P community can only work if there is a bigger global upload bandwidth (for more details, see [1]).

[a] Even if in most cases, there is no speed limit for upload by default, when installing the P2P client-server, for evident reasons: by this way, clients can download your files, if needed, as fast as you can *really* upload.

[b] Overwhelmed servers are a big percentage of P2P's servers, except when P2P servers are sharing uninteresting (for other users) files which are not much desired by the community.

Implications (continued)

The principle of P2P is to have a list of files (which is better to have one hand on), previously downloaded or not, which contains files that can be uploaded to other P2P's users. The aim of this paper is neither to show P2P's well-recognized interest, nor to make future predictions about their lifetime, but to show

1. that P2P has to be a community affair. If not, it cannot work,
2. and that making efforts to avoid P2P clients from litteraly consuming the whole bandwidth can be interesting.

Clearly, making P2P is as brilliant as well as Blizzard Entertainment uses it for its game, named "World of Warcraft"'s updates.

We could sum up these concepts in saying that *a compromise has to be found between an equal resources' sharing (as in any community) and a comfortable surfing.* Many solutions are available. For example, P2P programs could lower their download and upload speeds when the user's browser would be requesting pieces of information from the World-Wide Web (WWW). Diminishing the number of connections they habitually manage would not be appreciable to the user, as it would result in low performance, except if priorities were correctly managed.

Results

In traditional P2P sessions, and, for an average Internet connection, the number n_d of downloads is often limited (either by the user, or by the client) to $10 - 20$. In the following results, $n_d = 15$.

The Internet connection's parameters are defined as follows:

1. d_s is the *maximum theoretical download speed*,
2. u_s is the *maximum theoretical upload speed*,
3. $r := \frac{d_s}{u_s}$ is the *theoretical ratio download/upload*.

We use here a connection having $d_s = 4096\,\text{Kbps}$, $u_s = 256\,\text{Kbps}$, $r = 16$, from Belgium. No other application susceptible of Internet use is launched. This connection can be qualified of "cheap," as it costs 34 € by month.

Two different P2P *clients* are used: µTorrent (for BitTorrent protocol), and eMule (for eD2K protocol). For both, ports are open (resulting in "Network OK" and "High ID" respectively). Both clients have the same parameters.

It is well-established that files downloading on eD2K protocol is often slower during a while (depending on download files' availability) after the launch of the eD2K's client (*i.e.* BitTorrent protocol is faster at the beginning). This was ignored: chronometers were launched at the same time for both clients. As eMule takes 3 s of more than µTorrent to be ready, measures were adapted.

It must be taken into consideration that the measures rely on various factors that cannot be simultaneously controled, such as only one is modified at a time, in a purely scientific way. Thus, despite of the attention given to these experiments and measures, results can sometimes be unprecise.

We now give Time To Access Web Pages (TTAWP) when only one client is launched and when both clients are launched, depending on Time From Client Launch (TFCL), denoted by t_l. The data is plotted at Figure 1.

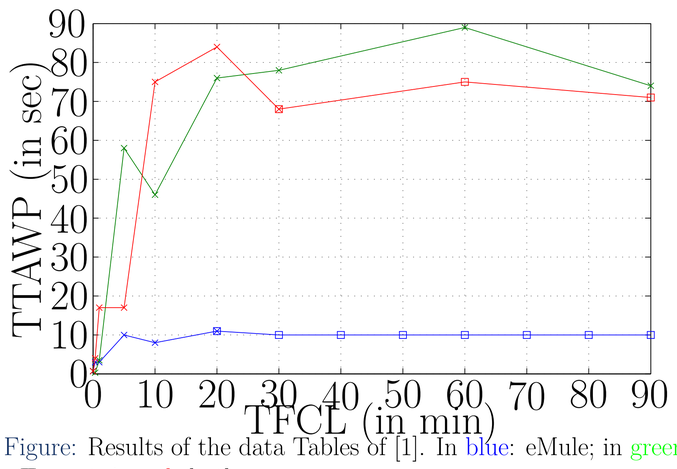

Figure: Results of the data Tables of [1]. In blue: eMule; in green: µTorrent; in red: both.

As the human reaction time equals here 1 s, and that it can vary with a term $\pm\Delta$, $\Delta = 0.5\,\text{s}$, the biggest difference between a real time t_r and the measure of this same time t_m can be up to $1 + 0.5 = 1.5$ seconds; that is, 1.5 s.

Proposition

The best way to surf on the Internet when downloading files *via* P2P would be to confer to P2P's clients the ability to auto-modify their settings, according to the Internet's browser's demands. P2P's clients could hear the potential Internet's browser's demands (e.g. to launch a page, or to quit) so that they modify their settings in an automatic way (e.g. to lower the number of connections, or to increase it).

Lowering *drastically* the number of connections would have a bad influence on the P2P's client's behaviour: it would be unuseful, and disturbating, as it would not download well, and would still use bandwidth which could be useful for web surfing. Clients for eD2K often keep numerous connections with other P2P's servers, even if they are very far in the Queue. It should be avoided. For example, constantly establishing numerous connections with lots of potential uploaders where your client is only in the end of their upload queue, has no interest. However, these parameters are not often taken into account, as the developers of P2P clients do not imagine either Internet connections with such a low bandwidth, or that one could dare surfing when P2P'ing.

Lowering the download speed would be uninteresting, whatever the percentage of lowering:

1. If the download speed is lowered drastically, the P2P's client has no interest,
2. If the download speed is not lowered drastically, the P2P's client still needs to send sufficiently to contribute to the community's wellbeing.

Lowering the upload speed would be interesting, but a few remarks have to be done:

1. If it is lowered too much (using a percentage to compare with the past upload speed), and that everybody does the same, the community will crash, as exposed previously,
2. If it is slightly lowered, it will anyway have an impact on the community, but it is not a problem if the DL speed is also lowered: people will share less, but download less, and surf more when P2P'ing.

It must be taken into account that the upload speed is often the lowest speed between DL and UL in commercial Internet solutions. As a result, even if the P2P's client does neither download too much, nor establish too much connections, the use of a big percentage (e.g. $> 70\%$) of the upload speed makes web surfing very slow.

Thus, when modifying parameters of his P2P's clients, one has to think constantly about:

1. What he wants to receive from P2P, and thus
2. What he wants to share with P2P, and thus
3. How to find a compromise between surfing as much as it is needed, and P2P'ing as much as he wants.

This is the only solution to follow until the P2P clients developers modify the in-depth foundations of their programs. We hope it will be done soon, for everybody's sake.

References

[1] L. Merciadri, *P2P Implications on Web Surfing.* 2009.

[2] Schollmeier, Rüdiger, *A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications*, Proceedings of the First International Conference on Peer-to-Peer Computing, (2002).