Playing with the Bandwidth Conservation Law

Farid Benbadis, Fabien Mathieu, Nidhi Hegde, Diego Perino

Playing with the Bandwidth Conservation Law

Farid Benbadis Fabien Mathieu Nidhi Hegde Diego Perino

Orange Labs,
38–40 rue du général Leclerc,
92794 Issy-les-Moulineaux, France

Abstract

We investigate performance bounds of P2P systems by application of the law of bandwidth conservation. This approach is quite general and allows us to consider various sharing systems such as fixed-rate streaming, VoD-type streaming, and elastic file sharing. Starting from a general law of bandwidth conservation, we consider several specific cases that apply to various P2P systems. For dynamic systems with a stationary arrival process, we show that simple seeding policies result in regimes where the download rates are arbitrarily fast. We consider a case with equal download rate among all peers as well as cases where the download rate is a function of upload rates, inspired by BitTorrent's Tit-for-Tat policy. In particular, we show that the sustainable proportion of free-riders is closely related to the Tit-for-Tat parameter.

1 Introduction

The peer-to-peer (P2P) paradigm has become widely popular in the last decade and has deeply impacted on how the Internet is perceived. Nodes are no longer simple customers, but actively participate in the information diffusion process by playing the role of clients and servers at the same time. In that way, the available resources are increased with respect to the traditional client-server approach.

The design of P2P systems aims to exploit the additional resources as best possible in order to improve scalability. At the same time, desired properties such as high service availability, fault tolerance, and low maintenance costs are design challenges of every P2P system.

Internet measurements show that P2P traffic represents up to 90% of global traffic, where BitTorrent [2] appears to be the most popular P2P system. Indeed the P2P paradigm goes far beyond file-sharing applications: other kinds of systems have recently increased in popularity like video streaming (e.g. Joost [11]) and VoIP (e.g. Skype [20]).

A recent research trend is the use of the P2P paradigm to improve the performance of a centralized service. Such an approach is called peer-assisted and is typically used to improve scalability of Internet Service Providers (ISPs) services ([5],[21]). This idea is enforced by the fact that ISPs have complete control on users' dedicated equipment (e.g. set-top-boxes) and they do not need to deal with unpredictable node resources and behaviors.

In any peer-to-peer or peer-assisted system, the sum of resources offered by peers at any time, is equal to the sum of resources they consume. In particular, the sum of data uploaded is equal to the sum of data downloaded (up to latencies shifts): this is the bandwidth conservation law, which is very similar to Kirchhoff's Current Law.

Contribution In this paper, we use the bandwidth conservation law to provide a unified model that can describe the performance of most bandwidth-consuming applications including, but not limited to, live streaming, video-on-demand and file-sharing.

Rather than focusing on a detailed model, we use the bandwidth conservation law as a point of departure, and derive explicit theoretical bounds on achievable performance. We further refine the bounds for specific cases. Specifically, under the assumption of flat allocation (of upload rates), we describe the leverage effect that can be used to provide rates greater than the average peer upload capacity.

We also provide a simple incentive model inspired by BitTorrent's Tit-for-Tat policy, and show that, with any kind of continuous arrivals/departures process, the system can handle a given percentage of free-riders and can enter an overprovisioning state where download performance can be arbitrarily great.

We derive conditions on system parameters such as the number of external servers, the required upload rates, the required Tit-for-Tat parameter, etc. These conditions can serve as guidelines for an ISP providing a peer-assisted service or for users tuning their application settings.

Our results serve as guidelines when considering the performance of real systems. The aim of the paper is not to propose explicit protocols but rather to provide theoretical bounds on what can be achieved in P2P systems, from the bandwidth-budget perspective.

Related work We now summarize some previous studies based on the conservation law in P2P systems. Yang and de Veciana [22] study the service capacity of a file sharing peer-to-peer system in both transient and steady state through a branching process model and a Markov chain model. Qiu and Srikant [18] study the steady-state network performance through a simple fluid model of a BitTorrent-like network. This approach has been recently used by Parvez *et al.* [15] for considering the impact of the piece selection scheme when streaming a video-on-demand.

We also focus on a steady-state system analysis and we extend these two works by considering various kinds of applications and by introducing the presence of external servers in the system. Moreover we consider arbitrary peer upload capacity distributions and different ways to distribute the total system capacity between peers.

The free-riders problem in BitTorrent has been considered by Yu *et al.* [23]. They propose dynamic resource allocation that depends on the nature of peers (free-riders or not), and show in a bimodal case that BitTorrent's built-in mechanisms can effectively handle some free-riders.

We generalize the bound on the number of free-riders a BitTorrent-like system is able to handle by considering arbitrary peer upload capacity distributions.

Liu *et al.* study [12] the performance bounds of a peer-assisted live streaming system and exhibit trade-offs between different system parameters, namely tree depth, server load, and degree. They focus on bounds of the aforementioned metrics by assuming the system is feasible, while we derive explicit conditions on the feasibility of the system.

Roadmap We introduce in Section 2 the underlying model that we use to express the conservation law. We also specify the various bandwidth requirements and performance evaluation metrics for the three main types of bandwidth-consuming applications: fixed rate (e.g. live streaming), required rate (e.g. Video-on-Demand), elastic rate (e.g. file-sharing). Then we consider in Section 3 the case of a fixed-size population. Section 4 studies the behavior when peers follow random arrival/departure processes, while we incorporate in Section 5 a Tit-for-Tat policy, for which numerical performance measures are proposed. Finally, Section 6 concludes the paper.

2 Model

We consider a hybrid P2P system where users collaborate to obtain a given service (i.e. file sharing, VoD, live streaming, ...). We classify users into three main categories:

Leechers The term is inspired by BitTorrent vocabulary and refers to users that are actually using the service. For example, users downloading content in a file sharing application or viewing a movie in a VoD/live streaming application.

Seeders This term is also inspired by BitTorrent vocabulary and indicates users that are not using the service, but are only providing resources to the system. For instance, peers sharing a completed file in a BitTorrent session or idle Joost nodes.

Servers Some external servers devoted to the service. There is no formal difference between a server and a seeder, but we distinguish the two kinds because their characteristics may be different (for instance servers may be less volatile and have more upload capacity than regular seeders).

We denote L, S and E to be the set of leechers, seeders and servers respectively. The number of leechers (resp. seeders) in the system, denoted by N_L (resp. N_S), may vary over time while we can consider the number of servers (represented as N_E) to be constant.

Every peer n in L, S, or E has an upload capacity u_n devoted to the service and every leecher l achieves a download rate of $d(l)$. The download capacity of leechers may be limited by $d_{\max}$, which we assume constant. In the bandwidth budget we only consider the effective data transfer and we do not take into account control messages or other overhead. Notation is summarized in Table 1.

2.1 Global bandwidth conservation

The bandwidth conservation law for our hybrid P2P system can be expressed as:

$$\sum_{l \in L} d(l) \leq \min(N_L d_{\max}, U_L + U_S + U_E), \qquad (1)$$

where $U_X = \sum_{n \in X} u_n$ indicates the total upload capacity of peers in set X.

Equation (1) states leechers cannot download faster than the sum of upload capacities of leechers, seeders and servers.

Note, that Equation (1) is only valid for unicast exchanges. If some peers have multicast or broadcast capabilities, the equation should be re-written while taking the leverage effect of multicast/broadcast into account. In particular, network coding at a network level using multicast, does not fit the model (but overlay network coding does). For simplicity we also neglect latency timeshifts.

2.2 Performance evaluation

The quality of service perceived by a leecher l is related to its download rate $d(l)$ and to the considered application. In the following, we focus on three applications representative of the most bandwidth-consuming ones:

Fixed rate Applications where the content is generated at a constant rate r and becomes available at a small subset of servers or seeders on the fly. The content has strict time requirements and should thus be distributed as fast as possible, so the buffer must be small. A measurement of quality of service is continuity in the content download: the download rate $d(l)$ of a leecher l should be equal to the current content generation rate r at all time. An example of such kind of applications is live streaming.

Required rate Applications where the content is available at a subset of nodes and should be downloaded at least at a given rate r. Once again the quality of service can be measured as continuity in the content download. However, in this case, $d(l)$ can also be greater than r because the content is already completely available and it is possible to store or cache it. A typical example of this kind of application is VoD streaming. The extra-bandwidth, besides facilitating the playback continuity, can also be used to provide advanced VCR functionality (Fast forwarding, Chapter jumping, etc.) [1].

Elastic rate Applications where the content is available at a subset of nodes but has no restrictions on download time or rate. A leecher l should try to maximize its download rate $d(l)$ in order to minimize its download time, which is the only metric to evaluate the quality of service perceived by a leecher. An example of such kind of applications is file sharing.

2.3 Bandwidth dispersion

In this section we investigate some possible reasons for an under-use of the available upload bandwidth, leading to a strict inequality in (1).

Content starvation Requested content may be totally or partially unavailable for download, leading to the incomplete utilization of available bandwidth. For instance, this is the case in the early phase of the diffusion of a file in BitTorrent, which is called *flashcrowd*. During the flashcrowd phase, the bottleneck is not the overall upload capacity, but the upload capacity of the initial seeder, which has to inject the first copy of the file in the system [17, 14].

Unnecessary bandwidth The service may not need all the available bandwidth (e.g. in fixed-rate scenarios). Note, that this scenario is very similar to content starvation, however, in a different context.

Download bandwidth constraints In some scenarios (for instance the over-seeding scenario described in Section 4.3), the sum of the physical download capacities of peers in L may be lower than the available upload bandwidth.

Implicit overhead In our model we do not take overhead into account. However, overhead at different network layers (TCP acknowledgments, overlay control messages, chunk collisions and retransmissions,...) may implicitly consume a part of the available bandwidth.

Hidden diffusion mechanisms The diffusion to leechers may require side-replication. Consider for instance the fixed-rate scenario described in Figure 1. A server $e \in E$ generates a content at a fixed rate r. Three leechers l_1, l_2 and l_3 want to receive the content, downloading it from the server e with the help of two seeders s_1 and s_2. The server e has upload capacity r while other peers have upload $\frac{r}{2}$. Notice that e and the leechers have a total upload capacity of $\frac{5}{2}r$, which is not enough (a minimal total bandwidth of $3r$ is required). A solution, shown in Figure 1, is to split the stream into three substreams r_1, r_2 and r_3 with upload rate of $\frac{r}{4}$, $\frac{r}{4}$ and $\frac{r}{2}$ respectively. Using s_1 and s_2 as re-transmitters for r_1 and r_2, r can be streamed to all leechers, with an upload cost of $\frac{7}{2}r$. In this example, the diffusion mechanism induces an upload overhead of $\frac{r}{2}$, which corresponds to transfers towards seeders. It can be shown that the minimal feasible overhead in this example is $\frac{r}{4}$, which can be obtained by splitting the stream into five substreams.

Non-optimal allocation The allocation protocol may fail to find an efficient allocation between uploaders and downloaders even if such an allocation exists.

We assume content availability is not an issue, and that the effect of implicit overhead and hidden diffusion mechanisms is negligible. Regarding the latter, it can be shown that the relative overhead can be arbitrarily small if N_S, N_L and the number of substreams are high enough [21]. Therefore, Equation 1 will be an equality unless bandwidth is over-provisioned or the allocation scheme is non-optimal.

2.4 Fluid model assumption

In the rest of the paper, we consider a steady-state fluid model: there is a large number of leechers $N_L >> 1$ and the download $d(l)$ of a leecher l is time independent.

Under the fluid model assumption, it is convenient to express the upload capacities of nodes belonging to a set X as a probability distribution function $p_X(u)$. Then the total and average bandwidth U_X and $\bar{u_X}$ can be defined as:

$$U_X = N_X \bar{u}_X = N_X \int u p_X(u) du. \qquad (2)$$

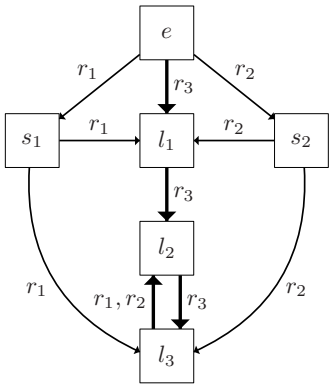


Figure 1. Example of hidden diffusion mechanism. The three leechers need auxiliary seeders to get the service from e.

Table 1. Table of notation

u_n	Available upload bandwidth of peer n
$d(l)$	Download rate of leecher l
r	Stream rate (if any)
k	File size (if any)
λ	Arrival intensity
$p(u)$	Upload bandwidth distribution
N_L (resp. N_S)	Number of leechers (resp. seeders)
T_L (resp. T_S)	Leecher (resp. seeder) time
U_X	Total upload capacity of population X
N_0	Normalized capacity of E ($N_0 = \frac{U_E}{r}$)
α	average upload/required rate ratio
β	Seeders/Leechers ratio
γ	Tit-for-Tat ratio

In the rest of this paper, we consider Equation (1) for different scenarios (fixed population, arrival/departure processes, flat/Tit-for-Tat bandwidth allocation).

3 Fixed population

In this section, we assume that U_E, N_L, p_L, N_S, and p_S are known. This is suitable for systems where precise statistics can be obtained.

Under the assumptions made in the previous section, the only difference between protocols, from a bandwidth point of view, is the way available bandwidth is allocated to the leechers. For now, we only consider the uniform (flat) bandwidth allocation, which means that the entire available bandwidth is equally split among all leechers (d is the same for all leechers). Other types of allocations will be considered later in the paper. Perfect flat allocation is achieved, for instance, when each uploader splits its own bandwidth among all leechers. However, it is more realistic to assume that each uploader chooses at random a few peers to whom it sends content. This results in a good approximation of a flat allocation.

3.1 Download performance

By applying Equation (1) with uniform allocation, we get:

$$N_L d = \min\left(N_L d_{\max}, N_L \bar{u_L} + N_S \bar{u_S} + U_E\right),$$

which leads to the following

$$d = \min(d_{\max}, \bar{u_L} + \beta \bar{u_S} + \frac{U_E}{N_L}), \text{ with } \beta = \frac{N_S}{N_L}. \quad (3)$$

For a given application, it is easy to compute d using Equation (3), and check if it fits the desired requirements.

3.2 Achievable rate

A dual problem is to consider a target rate $0 < r < d_{\max}$ and to describe the conditions under which this rate can be achieved. This approach is suitable to determine the feasibility of live broadcast services (live streaming, as PPLive [10]), or video-on-demand services (as Joost [11] or the Push-to-Peer system [21]).

If we denote α_L (resp. α_S) as the ratio $\bar{u}_L/r$ (resp. $\bar{u}_S/r$), and $N_0 = \frac{U_E}{r}$ as the maximum number of clients that the sources can serve on their own, then from (3) we write the following feasibility condition for rate r:

$$\alpha_L + \beta\alpha_S + \frac{N_0}{N_L} \geq 1. \quad (4)$$

We distinguish two situations: $\alpha_L + \beta\alpha_S \geq 1$ or $\alpha_L + \beta\alpha_S < 1$

If $\alpha_L + \beta\alpha_S \geq 1$, then the target rate r can be achieved for any source capacity and any number of leechers (the number of seeders must grow accordingly in order to keep β constant). We say the corresponding systems are scalable because the number of leechers they can handle is unbounded. This includes the case $\alpha_L \geq 1$ (leechers possess the necessary bandwidth by themselves, and do not need seeders or sources). Note that some live streaming [4] and video on demand [3] solutions are already available for $\alpha_L \geq 1 + \epsilon$.

In the special case where $\alpha_S = \alpha_L := \alpha$ (this happens for instance if seeders and leechers have the same upload distribution), then the scalability condition simplifies to $\alpha \geq \frac{1}{1+\beta}$: the average relative upload capacity must be greater than $a := \frac{1}{1+\beta}$. We call a the *activity* of the system ($a = \frac{N_L}{N_L+N_S}$ represents the proportion of leechers in the seeders/leechers population), and the scalability condition is simply:

$$\alpha \geq a. \quad (5)$$

On the other hand, if $\alpha_L + \beta\alpha_S < 1$, then the feasibility condition becomes

$$N_L \leq \frac{N_0}{1 - \alpha_L + \beta\alpha_S}. \quad (6)$$

We say the corresponding systems are not scalable because the number of leechers they can serve is bounded. However, one can notice that the source capacity, N_0, is leveraged by a factor $\frac{1}{1-\alpha_L+\beta\alpha_S}$. Therefore there is still an interest for such systems: by using P2P techniques, a content provider needs less of its own resources to feed a given number of clients.

3.3 Geographical smoothing

As stated by Equation (5), the relative required bandwidth in a scalable system must be greater than the activity. To illustrate this we consider Figure 2(a) which represents the activity rate within the Orange France digital television over IP (IPTV) network. Measures available here are based on data collected from February 4 to February 10 2008 and aggregated on a typical 24-hour day. The activity a is the ratio between the number of users that are actually watching TV and the total number subscribers of the service. From this figure, we see that the maximum activity rate is 53%, giving a lower bound for the relative upload bandwidth needed for providing this IPTV service on a P2P basis.

However, this activity rate exceeds 50% during only 2 hours per day, while the average activity in our example is only 36%. A natural question is: can we lower the required relative capacity from the maximum to the average rate?

One solution would be to proactively load the content during hours of low activity. Such a solution requires, unfortunately, to have an a priori knowledge of which content will be required, and is particularly inapplicable to live streaming content delivery.

On the other hand, if the service is proposed at a worldwide scale, then we have a natural smoothing of the activity because peak hours do not occur simultaneously across time zones. Thus, when demand is highest in Europe (between $8:30$ pm and $10:30$ pm), the service may use seeders from other regions to support the demand. Formally, if $a(t)$ designates local activity, which we suppose independent from the area, and P the distribution of users by time zone, then the overall activity A is the convolution of a by P: $A(t) = \sum_f a(t-f) \times P(f)$.

Figure 2(b) gives the global distribution of broadband users by time zone (data available at `http://www.internetworldstats.com/dsl.htm`). Assuming that P follows this distribution, the overall activity is shown in Figure 2(c). As convolution always smoothes the original curves, the maximum activity (and hence the relative required upload capacity) is now less than 39%, which is close to the minimum possible value (36%), and much better than the 53% observed at a local level.

The geographical smoothing allows a non-negligible gain, but it should be balanced with the fact that it uses (physically) distant links in order to lower the required activity. This may indeed lead to high latency and overloaded transcontinental links, which goes contrary to the current trends trying to improve locality of data exchange. We leave this question for a future study.

3.4 Case study: understanding Joost

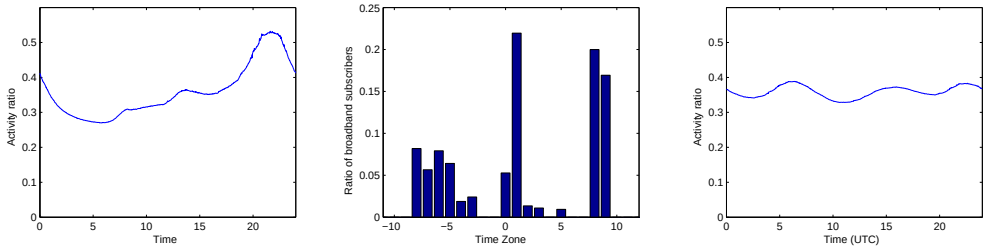
Joost [11] is a P2P video-on-demand service. Joost users install a dedicated client which behaves like a leecher (downloading and uploading content) when a user is watching a video. An idle Joost client can still upload content to others, acting like a seeder. From a measurement study performed by Hall, Piemonte and Weyant [9], it appears that Joost uses 700 kbps downstream when leeching, and in average 120 kbps upstream when leeching or seeding, leading to a relative upload capacity $\alpha \approx \frac{1}{6}$. In fact, the study indicates that about two third of the stream comes from dedicated servers. From (4), we deduce that β should be approximately 1 (there are about as many seeders as there are leechers). Note that no correlation between geographical location and transferred data was observed, suggesting geographical smoothing. In any case, Joost's architecture was still non-scalable at the time of the study.

What would it take for Joost to be scalable? From Equation (5), we see that we need a higher relative upload α, a lower activity a, or a combination of both. Relying on results of Section 3.3, geographical smoothing may be tempting for Joost designers (it costs them nothing), but burdening for the core network. The only other way to lower the activity is to enforce the seeding behavior (it is not in the interest of a VoD service to constrain the leechers' behavior). This can be done by proposing strong seeding incentives (for instance reduced commercials).

Otherwise, with the current measured $\beta \approx 1$ (or equivalently, $a \approx \frac{1}{2}$), the required α is at least $\frac{1}{2}$, corresponding to 350 kbps upstream dedicated to Joost. This represents a large bandwidth for today's DSL connections, which generally offer 1 Mbps upstream. The scalability upload cost may be easier with the new optical fiber connections, and perhaps the designers of Joost hope that the development of their service will coincide with an increase in access bandwidth.

4 Dynamic population

Following the approach proposed by Qiu and Srikant [18], we now consider that N_L and N_S are



(a) Activity during a typical day. (b) Broadband subscribers distribution (c) Worldwide activity (extrapolation).

Figure 2. Local activity and worldwide extrapolation during a typical day.

no longer fixed, but fluctuate according to a random arrival/departure processes. Formally, we suppose that peers join the system as leechers according to a process of intensity λ. The upload capacity distribution of the newcomers is indicated as $p(u)$. A given leecher l remains in leecher state for $T_L(l)$ time units before it becomes a seeder. A seeder s provides generous resources to the service for $T_S(s)$ time units, then it leaves the system.

Our approach is more general than the one proposed by Qiu and Srikant because we consider arbitrary upload distributions while they only consider one class with homogeneous upload capacity. As the systems we describe are more complex, we only consider in this paper the steady-state behavior of the system. Therefore the arrival process to all states (leechers or seeders) have the same intensity λ.

The leecher time T_L may be related to the peers' characteristics. Since the only difference between peers in our model is their upload bandwidth, T_L is related to the upload rate u so we can express T_L as a function of u i.e. $T_L(u)$. Following the same reasoning, we assume that T_S is also a function of the upload rate u i.e. $T_S(u)$.

From Little's Law we can derive the expected number of peers in set X in the steady state as:

$$\bar{N_x} = \lambda \bar{T_X} = \lambda \int T_X(u)p(u)du. \tag{7}$$

Note that our fluid model assumption is only valid for $N_X \gg 1$, which corresponds to $\lambda \bar{T_X} \gg 1$ according to Equation (7): the interarrival time should be small with respect to the expected leecher and seeder time.

The upload capacity distribution p_X of set X can also be easily deduced from the upload arrival distribution p and T_X: $p_X(u) = \frac{T_X(u)p(u)}{\bar{T_X}}$. p_X may be a sub-probability if T_X is zero for some values of u, but this issue is easily circumvented by setting the missing probability in u to 0 (peers that do not stay do not contribute to the system). It follows that Equation (2) can be rewritten as

$$U_X = \bar{N}_X \bar{u}_X = \lambda \int u T_X(u)p(u)du. \tag{8}$$

4.1 Download performance

We still consider a flat allocation as in Section 3: d is identical for all leechers (and constant in a steady state). Starting from now, we also assume that the content is of size k. This assumption is suitable for elastic/required rate applications but also for fixed rate ones if the content length is known in advance. Then the expected leecher time, $T_L = \frac{k}{d}$, is constant. It follows that: $p_L = p$, $\bar{u}_L = \bar{u} := \int up(u)du$, and $\bar{N}_L = \frac{\lambda k}{d}$.

We can now derive the download rate d as in Section 3.1:

$$d = \min(d_{\max}, \bar{u} + d\frac{\bar{T}_S \bar{u}_S}{k} + d\frac{U_E}{\lambda k}) \tag{9}$$

One can distinguish two cases in Equation (9). If $\bar{T}_S \bar{u}_S + \frac{U_E}{\lambda} \geq k$, the system is in a overprovisioning state, which will be further described in Section 4.3. Otherwise, Equation (9) becomes

$$d = \min(d_{\max}, \frac{u}{1 - \frac{\bar{T}_S \bar{u}_S}{k} - \frac{U_E}{\lambda k}}) \tag{10}$$

For a given application, it is easily verifiable if it is overprovisioned, and d can be computed accordingly.

4.2 Achievable rate

Like in Section 3.2 we can consider the dual problem of the feasibility of a target rate $r \leq d_{\max}$. We see in Section 4.3 that any rate can be achieved if $\bar{T}_S \bar{u}_S + \frac{U_E}{\lambda} \geq k$. Otherwise the condition, according to (10), is

$$r \leq \frac{\bar{u}}{1 - \frac{\bar{T}_S \bar{u}_S}{k} - \frac{U_E}{\lambda k}} \tag{11}$$

which leads to

$$\lambda k \leq \frac{U_E}{1 - \alpha - \beta\alpha_S}, \text{ with } \begin{cases} \alpha = \frac{\bar{u}}{r}, \\ \alpha_S = \frac{\bar{u}_S}{r}, \\ \beta = \frac{r\bar{T}_S}{k}. \end{cases} \tag{12}$$

Equation (12) is the equivalent to Equation (6): corresponding systems are not scalable (there is a maximal intensity), but the service capacity U_E is leveraged by $\frac{1}{1-\alpha_L+\beta\alpha_S}$.

4.3 Overprovisioning condition

Theorem 1 presents a sufficient condition for achieving optimal download rate $d_{\max}$ (the condition is easily translated to fixed-rate scenarios by replacing $d_{\max}$ by r). This condition is not limited to flat allocation, but works for any efficient allocation scheme (an allocation scheme is efficient if either the download bandwidth or the upload bandwidth is saturated).

Theorem 1. *A sufficient condition for achieving $d_{\max}$ with any efficient bandwidth allocation scheme is*

$$\bar{T}_S \bar{u}_S + \frac{U_E}{\lambda} > k(1 - \frac{\bar{u}}{d_{\max}}). \tag{13}$$

Two immediate corollaries of Theorem 1 are:

- if $\bar{T}_S \bar{u}_S + \frac{U_E}{\lambda} \geq k$, then any finite rate can be achieved,
- if $\bar{T}_S \bar{u}_S \geq k$, then any finite rate can be achieved independently of λ and U_E.

Remark The last condition is obviously verified if the system verifies $T_S(u) \geq \frac{k}{\bar{u}}$, because we then have $\bar{T}_S \bar{u}_S \geq \frac{k}{\bar{u}} \int up(u)du = k$. In other words, the download is optimal if all peers seed at least the time needed to get the file at a speed corresponding to the newcomers' average upload capacity. If no peer uploads at speed 0, $T_S(u) \geq \frac{k}{u}$ is also a sufficient condition: we get $\bar{T}_S \bar{u}_S \geq k\frac{u}{u} \int p(u)du = k$. In that case, the required condition is that each peer seeds at least the time needed to get the file at its own upload capacity. More generally, for any $0 \leq \mu \leq 1$, $T_S(u) \geq \mu\frac{k}{\bar{u}} + (1-\mu)\frac{k}{u}$ is a sufficient condition in absence of free-riders.

Proof. In the steady state, leechers arrive and leave with the same intensity λ. As any leecher downloads a quantity k between its arrival and its departure, the fluid limit of the total bandwidth $\sum_{l \in L} d(l)$ used by L is equal to λk. As the allocation scheme is efficient, if $U_L + U_S + U_E > \lambda k$, then the download is necessarily optimal. In particular, we have:

- $U_L = \lambda \bar{T}_L \bar{u}_L = \lambda \int uT_L(u)p(u)du$. As the download is limited by $d_{\max}$, we have $\forall l \in L, T_L(l) \geq \frac{k}{d_{\max}}$. Thus we have $U_L \geq \lambda \int u\frac{k}{d_{\max}} p(u)du = \lambda\frac{k}{d_{\max}}\bar{u}$;
- $U_S = \lambda \bar{T}_S \bar{u}_S$.

It follows that $U_L + U_S + U_E \geq \lambda\frac{k}{d_{\max}}\bar{u} + \lambda\bar{T}_S\bar{u}_S + U_E$, so if Equation (13) is verified, the download is necessarily saturated. □

5 Non-Flat allocation: Tit-for-Tat

We now consider a system where peers may have different download rates. The non-flat rate allocation can be related to several factors like different download capacities, number of applications running on the same host, RTTs among peers and so on. We assume that the overprovisioning condition does not hold (otherwise Theorem 1 applies). In order to keep the illustration simple, we consider that $d_{\max}$ is never reached (otherwise $\min(d_{\max}, .)$ should be inserted in all following equations).

We focus in this section on Tit-for-Tat allocation schemes, implemented in many file-sharing and streaming applications [6, 7, 16]. Under these Tit-for-Tat schemes, leechers share their bandwidth preferentially with those from whom they download the most. Therefore the download rate d of a given peer is partially determined by the upload capacity u.

It is generally assumed that Tit-for-Tat is driven by a parameter γ, $0 \leq \gamma \leq 1$ [23, 8]. Each leecher shares a fraction γ of its upload bandwidth according to its download history. The remaining $1-\gamma$ follows a flat allocation, as for the sources and seeders.

It is difficult to completely describe the behavior of the Tit-for-Tat exchanges, especially if peers are dynamic [8]. However, under some assumptions, it is possible to apply the Dirac limit of the stratification model [8]. In the Dirac limit, the Tit-for-Tat upload of a peer is rewarded by a download of the same value. We can then propose a general formulation of the download rate $d(u)$, which generalizes:

$$d(u) = \gamma u + (1-\gamma)\bar{u}_L + \frac{\bar{T}_S}{\bar{T}_L}\bar{u}_S + \frac{U_E}{\lambda \bar{T}_L}. \tag{14}$$

This equation can be solved numerically through an iterative process. It is valid for elastic and required-rate applications (we leave its adaptation for fixed-rate applications for future work). For $\gamma = 0$, it is equivalent to Equation (9).

In the following we describe solutions of this equation for some simple scenarios.

5.1 Leechers only systems

In this section we analyze a system without external servers ($U_E = 0$), and where leechers leave as soon as they finish their download ($T_S = N_S = 0$). Such systems have been proved feasible, for instance in BitTorrent after the initial content injection phase: a swarm of leechers can then work without any seeder if the arrival intensity is high enough ([13, 14]).

5.1.1 General case

Without seeders or sources, Equation (14) implies that $T_L(u) = \frac{k}{\gamma u + (1-\gamma)\bar{u}_L}$: the service of a peer only depends on u and $\bar{u}_L$, so finding $\bar{u}_L$ solves the system.

Using Equations (7) and (8), we derive that $\bar{u}_L$ must be the solution to the following equation:

$$\int \frac{\bar{u}_L p(u)}{\gamma u + (1-\gamma)\bar{u}_L} du = \int \frac{up(u)}{\gamma u + (1-\gamma)\bar{u}_L} du \quad (15)$$

Equation (15) can be solved numerically for any value of γ, as long as a steady state exists (cf Section 5.2). For the limit values, the solution is more explicit:

- $\gamma = 0$ implies $\bar{u}_L = \int up(u)du = \bar{u}$. The mean upload rate of leechers corresponds to the mean upload rate of newcomers, and it is the common download speed.
- If $\gamma = 1$, then each peer downloads at its own upload rate so that $T_L(u) = \frac{k}{u}$. In this case $\bar{u}_L$ is the harmonic mean of the upload rates of new comers. In particular, $\bar{u_L} \leq \bar{u}$.

More generally, by increasing γ, $\bar{u}_L$ should decrease, and the mean leecher time $\bar{T}_L$ should increase: faster peers leave sooner and the service cannot exploit their upload capacities for a long time.

5.1.2 Numerical evaluations

We study the numerical resolution of Equation (15) for two different peer upload capacity distributions: Gnutella Users and Uniform. The former, reported in Figure 3, is derived from a measurement study of Gnutella users [19] while the latter is a uniform distribution on a $[0, u_{\max}]$. $u_{\max}$ is set such that both distribution have the same average bandwidth.

We analyze the leecher mean upload capacity $\bar{u}_L$ as a function of the Tit-for-Tat parameter γ by means on numerical evaluations of Equation (14).

The results are presented in Figure 4. We numerically verify the results stated above: $\bar{u}_L$ is the arithmetic mean for $\gamma = 0$ and the harmonic mean for $\gamma = 1$. Both curves decrease, because by increasing the Tit-for-Tat incentive mechanism, faster peers finish their download earlier and contribute to the system for shorter periods of time.

The curves also show that the uniform distribution is less affected by γ than the Gnutella suggestion. This may be due to the high dispersion of the Gnutella distribution. In any case, the conclusion is that the solution of Equation (15) is highly sensitive to the bandwidth distribution.

5.1.3 Bimodal case

If the bandwidth distribution is bimodal, with a proportion p_1 (resp. $p_2 = 1 - p_1$) of arriving peers having an upload capacity u_1 (resp. u_2), Equation (15) becomes a quadratic equation:

$$(1-\gamma)\bar{u}_L^2 + ((\gamma - p_1)u_1 + (\gamma - p_2)u_2)\bar{u_L} - \gamma u_1 u_2 = 0 \quad (16)$$

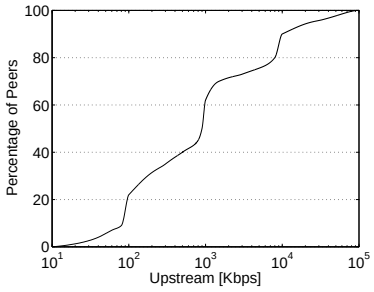


Figure 3. CDF of upload capacities of peers in the Gnutella system.

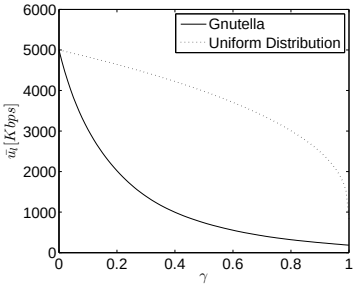


Figure 4. Average upload rate as a function of γ for two distinct upload distributions.

This equation can be solved easily for $0 \leq \gamma \leq 1$, $u_1 \geq 0$,$u_2 \geq 0$.

In the special case where $u_2 = 0$ (a part of the leechers, the *free-riders*, do not contribute at all to the system), the solutions of Equation (16) are $\bar{u}_L = \frac{p_1 - \gamma}{1-\gamma} u_1$ or $\bar{u}_L = 0$. In particular, for $p_1 \leq \gamma$ (or equivalently, $p_2 \geq (1-\gamma)$), $\bar{u}_L = 0$ is the only solution which makes sense. This critical value in the bimodal case has been already proved [23]. We propose now to extend this result to the general case.

5.2 Tolerance to free-riders

As stated above, a free-rider is a leecher that is not providing resources to the service (i.e. $u = 0$) but is just exploiting it. In the following, we consider a system where a proportion p_f of leechers are free-riders. The condition for a steady state to exist in presence of free-riders is given by the following theorem:

Theorem 2. *We assume that $\gamma u \leq d_{\max}$ for all u in the system. Then a necessary condition for the system stability is:*

$$p_f \leq (1-\gamma) + \gamma \left(\frac{\bar{T}_S \bar{u}_S}{k} + \frac{U_E}{\lambda k} \right) \quad (17)$$

Conversely, a steady state exists if Equation (17) *is strictly verified and if all bandwidth allocation schemes but the Tit-for-Tat are flat.*

Proof. In the steady state, if any, the total bandwidth used is λk, and the bandwidth used by free-riders is $p_f \lambda k$. For the steady state to exist, a corresponding upload capacity must be available for the free-riders. Equation (14) shows that regular leechers download at least at speed γu so that for them $T_L(u) \leq \frac{k}{\gamma}$. This leads to

$$U_L = \lambda \int_{u>0} uT_l(u)p(u)du \leq (1-p_f)\frac{\lambda k}{\gamma}. \quad (18)$$

Because of the Tit-for-Tat policy, the proportion of U_L allocated to free-riders cannot be more than $1-\gamma$. By comparing the needed and maximum available bandwidth for free-riders, one get the necessary condition

$$\begin{aligned} \lambda p_f k &\leq (1-\gamma)u_L + U_S + U_E \quad (19) \\ &\leq (1-\gamma)(1-p_f)\frac{\lambda k}{\gamma} \\ &\quad +\lambda \bar{T}_S \bar{u}_S + U_E, \quad (20) \end{aligned}$$

which leads to Equation (17).

Conversely, as p_f tends to its critical value, the proportion of free-riders tends towards 1 among the leechers population. If non-Tit-for-Tat schemes are flat, then all the generous bandwidth of the leechers (i.e. $(1-\gamma)U_L$) tends to be redistributed to free-riders, and as is the bandwidth of seeders. Thus the leechers' download speed tends to γu, and the bandwidth devoted to free-riders becomes arbitrarily close to $(1-p_f)(1-\gamma)\frac{\lambda k}{\gamma} + \lambda \int uT_s(u)p(u)du + U_E$. This retro-action mechanism ensures the existence and stability of a steady state if p_f is strictly below its critical value. □

If p_f is above its critical value, Theorem 2 implies that the system cannot be stable. In practice, the number of free-riders continuously grows because their arrival rate is too large with respect to their download rate. Note that, even under these conditions, the number of regular leechers is stable: they download at speed γu (but in practice, the overcrowded population of free-riders may affect the capacity for regular leechers to efficiently use their Tit-for-Tat schemes).

As a simple illustration, if we suppose $\gamma = 0.75$ (BitTorrent's default parameter) and $N_S = N_E = 0$, the system is able to tolerate up to 25% of free-riders.

5.3 Share Ratio policy

The Share Ratio (SR) is the ratio between the amount of data uploaded and the amount of data downloaded by a peer during its stay in the system. It can be considered as a metric to evaluate the contribution of a peer to a service with respect to the resources it exploited.

A common usage in BitTorrent communities is to impose a minimal share ratio to users, in order to improve the performance of the system. We analyze here the impact of such a Share Ratio policy by means of numerical evaluations of Equation (14). We use $k = 100Mb$, $\gamma = 0$ (flat allocation) or $\gamma = 0.75$ (BitTorrent's default setting), and the Gnutella Users distribution (Figure 3). The share ratio SR gives T_S: $T_S(u) = \max(0, \frac{SRk}{u} - T_L(u))$.

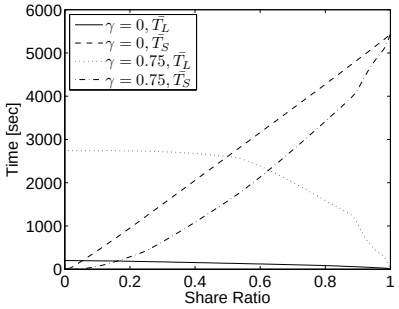


Figure 5. Leeching and seeding time as a function of share ratio for $\gamma = 0$ and $\gamma = 0.75$.

Figure 5 shows $\bar{T}_L$ and $\bar{T}_S$ as a function of the Share Ratio. Obviously, the share ratio increases the performance of the system by keeping U_S high. For $SR = 0$ leechers leave the system as soon as they finish their download as in Section 5.1. $SR = 1$ is a critical value corresponding to an overprovisioning state. Larger values of SR are not considered because they are not compatible with the existence of a steady state. As previously observed, the Tit-for-Tat mechanism increases $\bar{T}_L$.

Figure 6 reports T_L in average and for the three main bandwidth classes of the Gnutella distribution (we call these classes poor, normal and rich) when $\gamma = 0.75$. This figure explains in more detail the impact of γ and SR in scenarios where peers present highly heterogeneous upload capacities. We can observe that $\bar{T}_L$ is mostly affected by the performance of the poorer peers that achieve very high download time, even if SR can improve significantly their performance. On the other hand, richer peers rely on γ to achieve a high performance, and they are almost unaffected by SR.

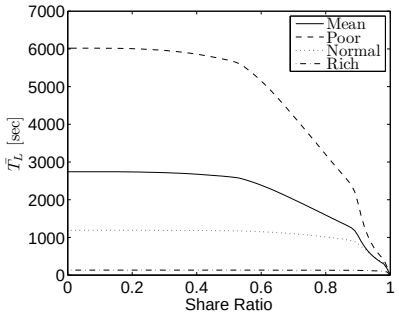


Figure 6. Leeching time (average and per-class) as a function of share ratio for $\gamma = 0.75$.

6 Conclusion

We have proposed in this paper a unified model, based on the bandwidth conservation law, which describes the performance of a large number of today's popular applications, such as live streaming, video on demand, and file sharing. Our model provides theoretical bounds on the achievable performance of such systems. We have described how, using P2P, a provider can reduce costs while increasing the capacity of its network. Under a steady state assumption, we showed an overprovisioning condition that guarantees an arbitrarily low download time.

In order to face free-riding users, we have also proposed an incentive model based on BitTorrent's Tit-For-Tat, and showed that this model, under a continuous arrival/departure process, is able to handle a certain ratio of free-riders.

Our model is indeed general enough to design incentive mechanisms, impose rate limitations or requirements, and specify conditions on other parameters, to allow not only for the proper tuning of current peer-to-peer or peer-assisted applications, but also for the construction of new peer-assisted services.

Acknowledgments This work was partially supported by the projects NAPA-WiNe (FP7-ICT-2007-1, grant 214412) and P2Pim@ges.

References

[1] S. Annapureddy, S. Guha, C. Gkantsidis, D. Gunawardena, and P. Rodriguez. Exploring VoD in P2P swarming systems. In *INFOCOM*, pages 2571–2575, 2007.

[2] BitTorrent. http://www.bittorrent.com/.

[3] Y. Boufkhad, F. Mathieu, F. de Montgolfier, D. Perino, and L. Viennot. Achievable catalog size in Peer-to-Peer Video-on-Demand systems. In *IPTS*, 2008.

[4] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: high-bandwidth multicast in cooperative environments. In *SOSP*, pages 298–313, New York, NY, USA, 2003. ACM.

[5] M. Cha, P. Rodriguez, S. Moon, and J. Crowcroft. On next-generation telco-managed P2P TV architectures. In *Proceedings of IPTPS*, 2008.

[6] B. Cohen. Incentives build robustness in bittorrent. In *P2P ECON*, 2003.

[7] A. Gai and L. Viennot. Incentive, resilience and load balancing in multicasting through clustered de bruijn overlay network (prefixstream). In *Proceedings of the 14th IEEE International Conference on Networks (ICON)*, volume 2, pages 1–6. IEEE Computer Society, September 2006.

[8] A.-T. Gai, F. Mathieu, F. de Montgolfier, and J. Reynier. Stratification in P2P networks: Application to bittorrent. In *ICDCS*, 2007.

[9] Y. J. Hall, P. Piemonte, and M. Weyant. Joost: A measurement study. Technical report, School of Computer Science, Carnegie-Mellon University, may 2007.

[10] X. Hei, C. Liang, J. Liang, Y. Liu, and K. W. Ross. Insights into pplive: A measurement study of a large-scale p2p iptv system. In *IPTV Workshop*, 2006.

[11] Joost. http://www.joost.com/.

[12] S. Liu, R. Zhang-Shen, W. Jiang, J. Rexford, and M. Chiang. Performance bounds of peer-assisted live streaming. In *Proceedings of Sigmetrics*, 2008.

[13] L. Massoulié and M. VojnoviĆ. Coupon replication systems. *SIGMETRICS Perform. Eval. Rev.*, 33(1):2–13, 2005.

[14] F. Mathieu and J. Reynier. Missing piece issue and upload strategies in flashcrowds and P2P-assisted filesharing. In *P2PSA*, 2006.

[15] N. Parvez, C. Williamson, A. Mahanti, and N. Carlsson. Analysis of bittorrent-like protocols for on-demand stored media streaming. In *Proceedings of the 2008 ACM SIGMETRICS International Conference*, pages 301–312, 2008.

[16] F. Pianese and D. Perino. Resource and locality awareness in an incentive-based p2p live streaming system. In *P2P-TV '07: Proceedings of the 2007 workshop on Peer-to-peer streaming and IP-TV*, pages 317–322, New York, NY, USA, 2007. ACM.

[17] J. A. Pouwelse, P. Garbacki, D. H. J. Epema, and H. J. Sips. The Bittorrent P2P file-sharing system: Measurements and analysis. In *4th Int'l Workshop on Peer-to-Peer Systems (IPTPS)*, Feb 2005.

[18] D. Qiu and R. Srikant. Modeling and performance analysis of bittorrent-like peer-to-peer networks. In *Proceedings of ACM SIGCOMM*, pages 367–378, New York, NY, USA, 2004. ACM.

[19] S. Saroiu, P. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking*, 2002.

[20] Skype. http://www.skype.com/.

[21] K. Suh, C. Diot, J. F. Kurose, L. Massoulié, C. Neumann, D. Towsley, and M. Varvello. Push-to-Peer Video-on-Demand system: design and evaluation. *IEEE JSAC*, 25(9), 2007.

[22] X. Yang and G. de Veciana. Service capacity of peer to peer networks. In *Proceedings of IEEE INFOCOM*, 2004.

[23] J. Yu, M. Li, F. Hong, and G. Xue. Free-riding analysis of bittorrent-like peer-to-peer networks. In *APSCC '06: Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing*, pages 534–538, Washington, DC, USA, 2006. IEEE Computer Society.