

System Modeling for Priority Schemes in Managed Peer-to-Peer Networks

Mario E. Rivero-Angeles¹, Izlian Yolanda Orea-Flores²

1 Instituto Politécnico Nacional (CIC-IPN),
Mexico

2 Instituto Politécnico Nacional, Telematics Department, UPIITA-IPN,
Mexico

{mriveroa, ioref}@ipn.mx

Abstract. In Peer-to-Peer (P2P) networks it is not uncommon to find users that have initiated a file transfer and decide to leave the system before the end of the download. This is a particularly harmful behavior due to the resources (bandwidth or energy) wasted in such an aborted process. This negative effect is amplified when the system's conditions are such that the peers are not downloading at the maximum capacity. This is because in these conditions, there are not enough peers uploading the file, and a part of the bandwidth is wasted on peers that do not share their resources to the network once they leave the system. In this paper, the case of managed P2P networks is of special interest. As such, the variability in the transfer rates, such as the download and upload rates, is expected to be much less than in the case of the unmanaged P2P network. The limit case, where these transfer rates are constant is considered as an alternative to the exponential distribution. Additionally, two priority schemes are presented for a BitTorrent-based P2P network where the peers that are more likely to remain longer in the system are served first.

Keywords. Managed peer to peer network, priority scheme, performance evaluation, transfer rates modeling, BitTorrent.

1 Introduction

A Peer-to-Peer (P2P) network is a distributed network with dynamic nodes, called peers. The peers in the system provide resources such as bandwidth, storage space, computing power, with the objective of exchange the data or of performing some collective task. In recent years, P2P systems have received a great amount of attention since

applications based on these types of networks are dominating a considerable part of the traffic in the Internet, as noted in [4, 5]. Specially for TV streaming applications (live or stored), as studied in [6] and [7]. Indeed, a big effort has been done to implement IPTV services which consume high amount of bandwidth [8, 9].

As such, a P2P architecture can be a good alternative for such services in future networks. One great advantage of a P2P system over conventional client/server applications resides in its robustness. This is because the data is no longer concentrated in one central entity such as a server; instead, they are distributed over the population of users.

Another advantage is scalability. This is due to the fact that all peers in the system share their resources and any new arriving peer, while increasing the demand, also increases the overall system's capacity.

One basic disadvantage is that P2P networks suffer from the total absence of control on content distribution, as they are running on PCs and are managed by the users themselves. Users' PCs are unmonitored, not secure, and almost impossible to control by a network provider (we exclude here the case of managed networks, that is, of networks belonging to some telco or ISP).

In this work, we focus on file distributing networks, and on the harmful effect caused by peers that enter the system and begin the file downloading process but leave before finishing it or just after finishing their own download. Hence,

resources can be assigned to a peer that is not staying long enough in the network to cooperate with the rest of the users, that is, leading to a waste.

We explore here a reward-based P2P network where the peers that have a high probability of staying in the system long enough to share the file are served first. This mechanism is in part encouraged by the results presented in [1], where it is shown that in a cooperation game, rewards are as effective as punishments for maintaining cooperation. Also, the use of rewards leads to higher total earnings.

Applying this philosophy to a P2P network, our priority scheme rewards the peers that have a high delay tolerance to download and share the file. On the other hand, the peers that waste the system's resources by leaving immediately after downloading their file or even before downloading the complete file are not punished in any way, they are just not rewarded. In fact, when the system has enough resources, the priority scheme does not affect the low tolerant peers at all. For this priority scheme, the BitTorrent application has been chosen.

As an additional contribution of this work, different transfer rate distributions are considered for the uploading and downloading bandwidths. The exponential model for the upload and download rates is a simple model that has the advantage of allowing a simple mathematical analysis. However, one big disadvantage is that it does not closely match a practical P2P network. This is because the exponential distribution has a very high variability. Hence, while it is a good approximation for arrival and leaving rates of nodes into the system, upload and download rates are not expected to have such a high variability. In some cases, these rates have been considered to be constant [14]- [17]. In this work, we are particularly interested on the limit case where the transfer rates are constant. This case can be used to model a managed P2P network where the communicating devices are provided by the service providers such as Internet and television services. In managed networks, the system's parameters are typically under tighter control than in unmanaged networks. Hence, even the rates of aborting the download or leaving the system after the download can also be

adjusted by the service providers. We only focus the discussion on the transfer rates and we leave the study of the impact of the departure rates on the system for a future work. Finally, the priority models are also studied for the constant transfer rates as compared to the exponential case.

For the priority scheme implementation and the study of the performance of the system with different bandwidth models, the BitTorrent application has been chosen.

1.1 BitTorrent

BitTorrent is a P2P application used to facilitate the download of popular files. The main idea is to divide the files into several pieces called *chunks*. To download a file, peers exchange these chunks following some rules.

The BitTorrent protocol differentiates two types of peers: *leeches*, which are peers that have a part of the file or not data at all, and *seeds*, which are those peers having downloaded the complete file and remaining in the system to share their resources. Both leeches and seeds cooperate to upload the file to other leeches. Whenever a peer joins the system with the objective of downloading the file, it contacts a particular node called *tracker* which has the complete list of peers that have part or the complete file. Then, the tracker returns a random list of potential peers that might share the file with the arriving peer. At this point, the downloading peer contacts the peers on the list and establishes which chunks it is willing to download from each peer it is connected with. The decision of which peers to upload to depends basically on how much chunks have downloaded the peers in question. Hence, the peers that have downloaded the most are the ones that have priority over the peers that do not share their chunks, discouraging the free riders (peers that only download but do not share their data).

1.2 Contributions

- This paper proposes two simple priority schemes to serve first the peers that are statistically more likely to remain in the system until the download of the file is completed, and

analyses the performances of both proposals. Priority schemes have not been the subject of significant research efforts so far, in this area. A simplified model of a P2P network is given, with the objective of understanding some aspects of the behavior of such a system. For implementation purposes, we chose the BitTorrent application since the tracker has access to all the necessary statistical information about the peers in the system. Specifically, the tracker can inform the rest of the peers about the average time that a leech remains in the network. We also study the conditions under which the priority schemes may be useful.

- The impact of different distributions for the transfer rates on the performance of the P2P system is analyzed. In particular, we are interested on the effect of the variability generated by the different rate distributions. Hence, the exponential distributions and the constant value models are considered. The mean and variance of the number of peers in the system as well as the downloading, waiting and life time of leeches are considered as performance metrics of the system.

The rest of the paper is organized as follows. Section II reviews the previous works related to mathematical modeling and incentives in P2P networks. Then, section III specifies the system model and assumptions, including the analysis of the system's conditions where the priority schemes are suited, the models used in this work and the simulation model. Following this, section IV presents the relevant numerical results in order to evaluate the impact on the distribution of the transfer rates and the priority based mechanisms proposed in section III. The article concludes with a summary of our conclusions and contributions.

2 Related Work

We built our priority model based on [2], and using also [3, 4, 13]. Unlike these works, we consider a two-population model. Additionally, a priority mechanism is proposed and studied by means of

a fluid simplification and a Markov chain. Fluid models as the one in [2], allow simple analytical discussions. We complete the analysis with a Markov chain model such as the one presented in [13] to get better insight into the performance of the system. In [12], two classes of users are considered, namely: high bandwidth users and low bandwidth users. This is done in order to approximate a real system where different users have different hardware characteristics. Unlike [12], we focus on the behavior of the different peers during the downloading procedure rather than the different bandwidth capacities that can be encountered in a real network. Moreover, an event simulator is implemented to study a managed P2P network where the transfer rates are considered to be constant. This model can no longer be studied using neither the Markov chain nor the Fluid model.

Additionally, unlike previous works, we consider two priority based schemes which are different from the general idea of incentives such as [10, 11]. While the incentives are used as means of influencing a certain behavior in the peers any time the peers are connected to the network, our proposal is used only when the resources in the system are scarce. Meanwhile, when resources are sufficient in the network, there is no need to apply the priority scheme. In other words, the priority schemes can be seen as a means of improving the performance of the system whenever the environment is adverse in terms of insufficient upload bandwidth. However, the incentive mechanisms can work in parallel with the proposed priority schemes in order to further improve the system's performance. This issue is left to further future analysis. The priority schemes are studied for both the exponential model and the constant model.

In [14]- [17], a constant value on the transfer rates is considered. In [14, 16] an arbitrary value is assumed while in [15] different values on the upload and download rates are considered according to different system's conditions. However, no further details on the nature of these rates nor a more detailed description is given. In contrast, in this work, we investigate the impact on the system when considering constant data rates (as the

works previously described) and random variables with exponential distribution as in [2] and [13].

3 System Models

First, we briefly describe the fluid process used in [2] for the single population model and then, the proposed two population and priority models. The system parameters used throughout the paper, unless specifically stated otherwise, are presented in Table 1 [2].

3.1 Single Population Models

The mathematical models presented in this section are a simplification of a BitTorrent-like network. Indeed, the real system is much more complex due to the numerous parameters and details involved in the file searching, node selection, and chunk selection procedures, among others. Specifically, the mathematical models assume that whenever a peer arrives, it is immediately served by the rest of the peers in the network, i.e., the total uploading bandwidth is uniformly distributed among all users in the system. Another important assumption is that all the peers in the system cooperate, or that all users have chunks that are of interest for the rest of the peers. This is a very important problematic that has been extensively studied before (for instance in [13], the effectiveness of the file sharing is studied with a parameter η that represents the probability that a peer is interested in the chunks of other peers, similar also to the work presented in [2]) and therefore we do not further study this issue. Finally, one more detail that is not represented in the mathematical models is the fact that the BitTorrent system only allows for a maximum of 4 active uploading connections for any peer. An additional peer is selected randomly in order to better distribute the file in the network under the *optimistic unchoking policy*. Next we describe the rest of the assumptions and system parameters.

New peers arrive to the system according to a Poisson process having rate λ and are labeled as leeches. All peers have the same uploading rate μ and the same downloading rate c , $c > \mu$. A single file download is considered. At any given time there is at least one seed in the system. All peers

have complete knowledge of the system, i.e., all peers know which chunks has any other peer and peers always cooperate to upload data if they have available bandwidth. As such, if the number of leeches and seeds is sufficiently high, all leeches download the file at the maximum download rate c . However, when there are not enough peers in the system, the leeches download at rate $\mu(x + y)$, where x is the instantaneous number of leeches and y is the instantaneous number of seeds. With respect to [2], we simplify the model by assuming that the efficiency η defined in that paper is 1. From the previous description, the evolution with time x and y satisfies:

$$x'(t) = \lambda - \theta x(t) - \min(cx(t), \mu[x(t) + y(t)]), \quad (1)$$

$$y'(t) = \min(cx(t), \mu[x(t) + y(t)]) - \gamma y(t). \quad (2)$$

Observe that in these equations, stated in [2], the condition $y(t) \geq 1$ is not considered. One way to do it is to denote by $y(t)$ the number of seeds in the system other than the fixed ones, at time t , and to replace $\min(cx(t), \mu[x(t) + y(t)])$ by $\min(cx(t), \mu[x(t) + y(t) + 1])$, but we kept the equations as in the original paper, for simplicity. Remember that these differential equations make sense for high population values.

In [2], this dynamical system is analyzed and its equilibrium point, denoted by (\bar{x}, \bar{y}) , is studied. The equilibrium is obtained by solving the system $\{x'(t) = 0, y'(t) = 0\}$ and by replacing $x(t)$ by \bar{x} and $y(t)$ by \bar{y} . The result is:

$$\bar{x} = \frac{\lambda}{\beta \left(1 + \frac{\theta}{\beta}\right)}, \quad \bar{y} = \frac{\lambda}{\gamma \left(1 + \frac{\theta}{\beta}\right)}, \quad (3)$$

where $\beta = (\max\{1/c, 1/\mu - 1/\gamma\})^{-1}$.

Consider now the evolution of the pair $W(t) = (L(t), S(t)) \geq (0, 1)$, where $L(t)$ and $S(t)$ are respectively the number of leeches and seeds at time t . With all the previous "exponential assumptions", $W(t)$ is a continuous time homogeneous Markov chain, with initial state $W(0) = (0, 1)$. Starting from state (n, m) , $n \geq 0$ and $m \geq 1$, the transition rates are as follows: λ to state $(n + 1, m)$, $n\theta$ to $(n - 1, m)$ (for $n \geq 1$), $\min(cn, \mu(n + m))$ to $(n - 1, m + 1)$ (for $n \geq 1$), and $(m - 1)\gamma$ to state $(n, m - 1)$ (for $m \geq 2$).

Table 1. Parameters setting

Parameter	Description	Value
λ	Arrival rate	0.04, 0.4, 4
c	Downloading bandwidth	0.002
$\mu (\mu \leq c)$	Uploading bandwidth	0.00125
γ	Departure rate for Seeds	0.001
θ	Departure rate for Leeches	0.001
F	File size	1

This Markov chain is analyzed using simulation. Concerning its stability, it is easy to see that it is always stable by seeing the model as a tandem of infinite server queues, the first one containing the leeches, the second one the seeds. The second one is a simple M/∞ queue. The first one is a more complex M/∞ queue, with a service rate depending on the state of the whole system. Stability follows by lower bounding the service rate at the first queue by $\mu L(t)$, for instance, and by observing that the resulting model is a Jackson network composed of two M/∞ queues, which is always stable.

For the priority scheme, it is important to identify the system's condition where the bandwidth is scarce due to the low number of peers. Then, numerical values of the number of seeds and leeches in equilibrium are obtained from both the Markov model and the fluid model. It can be seen from Fig. 1, that the Markov model (referred as *simulation*) achieves a very good match to the closed expressions presented in (3), when $cx < \mu(x + y)$. This condition of *abundance* (also referred in this document as *maximum capacity*), is achieved when there is a high number of leeches and seeds in the system, i.e, for θ and γ small and λ high. The values of c and μ are not varied since these parameters are usually defined in advance and in general do not depend on the user behavior.

In Fig. 1, we can see that when the departure rate of leeches is high, both the number of seeds and the number of leeches is low. This is because most of the peers that enter the system leaves in a small amount of time and there are almost no leeches that convert into seeds. Conversely, when the departure rate of seeds is high, the number of leeches does not decrease since there

are still other leeches in the system who share the file with the rest of the peers. The number of seeds does not decrease as drastically as in case of high values of θ , since there are still leeches becoming seeds. More importantly, when $\gamma \geq 0.004$, the system has a high probability to be in *penury* conditions. Hence, for the introduction of the priority scheme, the value $\gamma = 0.001$ is chosen to represent *abundance* conditions whereas the value $\gamma = 0.005$ is chosen to represent *penury* conditions.

3.2 Two Population Model

For the two population model, the behavior of two types of users is considered: cooperative leeches (also called *high tolerance leeches*) and defeat leeches (also called *low tolerance leeches*). For the former, they are peers that arrive to the system and usually stay throughout the entire download procedure. In other words, they have a high tolerance for download latency. Therefore, their departure rate (θ_c) is lower than the download rate at the maximum capacity, i.e., $c > \theta_c$. For the latter, they are peers that arrive to the system and have very little tolerance to download latency. Hence, even if their departure rate is lower than the download rate at the maximum capacity, it is just lower. In view of this, we consider the next set of values: $\theta_c = \theta$ and $\theta_d = 0.9c$. Cooperative leeches arrive with rate λ_c and defeat leeches arrive with rate λ_d .

According to this, the fluid model presented above is extended for the two types of leeches as

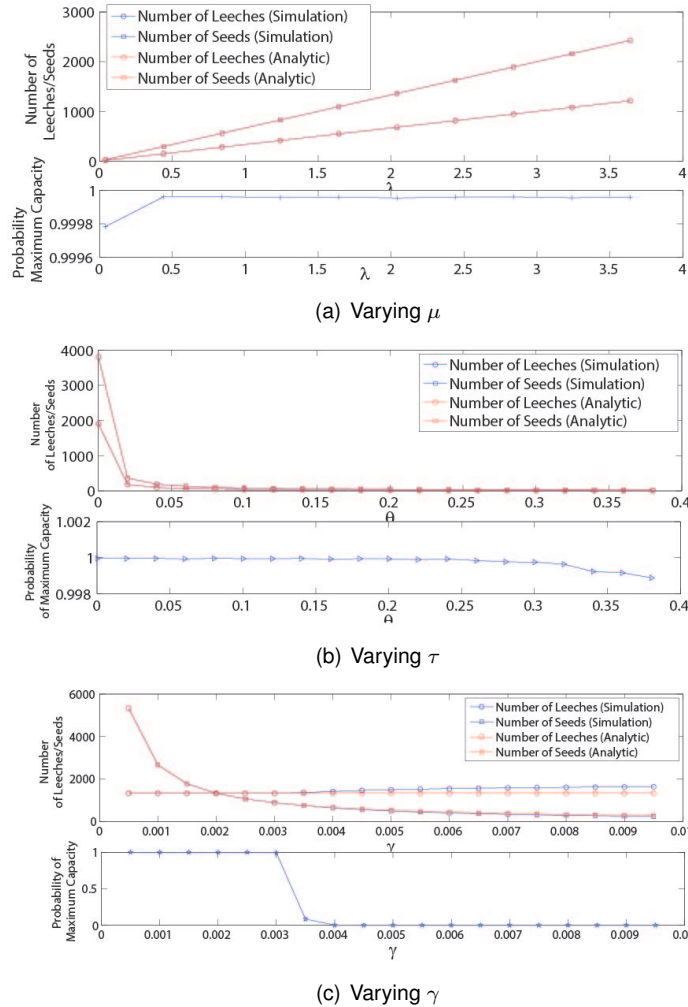


Fig. 1. Average Number of Leeches and Seeds for Different Arrival and Departure Rates

follows:

$$x'_c(t) = \lambda_c - \theta_c x_c(t) - u_c(t), \quad (4)$$

$$x'_d(t) = \lambda_d - \theta_d x_d(t) - u_d(t), \quad (5)$$

$$y'(t) = u_c(t) + u_d(t) - \gamma y(t). \quad (6)$$

with the additional notation $x(t) = x_c(t) + x_d(t)$, $n(t) = x(t) + y(t) + 1$, $u_c(t) = \min(cx_c(t), \mu n(t)x_c(t)/x(t))$, and $u_d(t) = \min(cx_d(t), \mu n(t)x_d(t)/x(t))$. In order to find the steady state values, the equilibrium values for the number of each class of leeches and seeds are calculated according to the same

procedure described in [2]:

$$\bar{x}_c = \frac{\lambda_c}{\beta \left(1 + \frac{\theta_c}{\beta}\right)}, \quad \bar{x}_d = \frac{\lambda_d}{\beta \left(1 + \frac{\theta_d}{\beta}\right)}, \quad (7)$$

$$\bar{y} = \frac{\beta}{\gamma} \bar{x}, \quad \text{with } \bar{x} = \bar{x}_c + \bar{x}_d. \quad (8)$$

The Markov chain presented in the previous subsection is also extended according to the two new 2-class model. The new model is the triple $(L_c(t), L_d(t), S(t))$ with the obvious meaning. The initial state is $(0, 0, 1)$. Starting from state (n_c, n_d, m) , the transition rates are: λ_c to state

$(n_c + 1, n_d, m)$ and λ_d to state $(n_c, n_d + 1, m)$; $n\theta_c$ to state $(n_c - 1, n_d, m)$ (for $n_c \geq 1$) and $n\theta_d$ to state $(n_c, n_d - 1, m)$ (for $n_d \geq 1$); $(m - 1)\gamma$ to state $(n_c, n_d, m - 1)$ (for $m \geq 2$); $\min\{cn_c, \mu(n_c + n_d + m)n_c/(n_c + n_d)\}$ to state $(n_c - 1, n_d, m + 1)$ (for $n_c \geq 1$) and $\min\{cn_d, \mu(n_c + n_d + m)n_d/(n_c + n_d)\}$ to state $(n_c, n_d - 1, m + 1)$ (for $n_d \geq 1$).

3.3 P2P Network Simulation at Peer level and Constant Time Rates

In this section, the event simulation model at *peer level* is discussed in detail. The main objective of this simulator is to study a managed P2P network where the transfer rates are constant. Additionally, it allows to verify the results derived from the mathematical models and to obtain different performance metrics in order to have more insight of a P2P system.

As such, the simulations are performed as close as possible to both the Markov and Fluid models but without some important assumptions. Specifically, the aforementioned models consider that all the peers, from the moment of the arrival of the peer to the moment of the peer departure, always cooperate to share the file in the system. Conversely, the simulation considers that the peers can only share the file once it has downloaded the file. A more detailed simulation, considered for future work would consider the sharing of the file once it has downloaded at least some chunks of the file.

Another important difference is that, the simulator considers that whenever the active peers are uploading the file to other peers, they upload at the maximum rate μ . Hence, it cannot serve more than one peer at the time. On the other hand, the mathematical models consider that whenever a new peer arrives, the rest of the active peers in the network immediately begin to share the file, hence any given peer can serve more than one peer simultaneously.

The pseudo-code of the event simulator is presented in Algorithm 1. Four types of events are considered: Arrival of a new peer, End of the download of the file, Departure of a Leech, and Departure of a Seed. Each arriving peer is assigned an individual and particular identifier

id. Hence, in the pseudo-code, the identifier of the peer that is being attended according to the different events is referred as *id*. And the identifier of the peer that is uploading the file is referred as *idupload*.

In the peer arrival event, the new peer looks for an idle peer, i.e., a peer that is not uploading the file to any other peer. Indeed, since a peer that shares the file uploads at the upload rate μ it cannot upload to any other peer. And the new arriving peer continues to look for other peers until the download capacity is fulfilled. For each peer that shares the file, it is marked as *busy* as an indicator that it cannot upload the file to another peer. When the file is downloaded, these uploading peers are marked as *idle* and they can now attend other peers. Also, the peer that finished the download is converted to a seed and is also marked as *idle*. Since a new peer programs its departure of the system, whenever an *End Download* event occurs, all the events related to the peer *id* have to be removed from the event list. This is also the case of the *Departure Leech* event. However, for the *Departure Seed* event, this is not necessary since a Seed only has this event on the event list. Finally, the *expo* instruction corresponds to the generation of an exponential distributed random variable.

An additional advantage of this event simulator is that it allows the use of different distributions for the time variables. In particular, we want to investigate the system when the transfer rates are constant. The interest of fixing the download and upload rates to constant values, is that it models one important aspect of managed P2P networks. As such, it is possible to evaluate the impact of the priority schemes on these systems. Indeed, in a managed P2P network, the service providers use their own devices which typically have the same upload and download rates. This is an important difference compared to the case of a more general P2P system, where users typically use their PCs or mobile devices (laptops, smart phones, etc.). It is clear that these devices have very different capabilities for both hardware and software. Also note that the Markov model and Fluid model are no longer valid in this case. Finally, the implementation of the constant model in the simulator is straightforward.

Algorithm 1: Pseudocode of the Event Simulator

```

while (Sim time ≤ 20000)
  case Peer Arrival:
  Num Leeches++;
  if(idle peers = true)
    Download = expo(c);
    while(Upload ≤ Download)
      Register idupload to peer id;
      peer idupload → busy;
      Upload + = expo(μ);
    end while
    Program "End Download" in Sim time +  $\frac{1}{U_{pload}}$ ;
  else
    peer id → waiting;
  end if
  T = expo( $\frac{1}{\beta}$ );
  Program "Departure Leech" in Sim time + T;
1 case End Download:
  Num Leeches--;
  Num Seeds++;
  Delete events of peer id;
  peer id → idle;
  peer idupload → idle;
  T = expo( $\frac{1}{\gamma}$ );
  Program "Departure Seed" in Sim time + T;
case Departure Leech:
  Num Leeches--;
  Delete peer id;
  Delete events of peer id;
case Departure Seed:
  if(Num Seeds > 1)
    Num Seeds--;
    Delete peer id;
  end if
end while

```

4 Priority Models

Two different priority schemes are proposed. In the first scheme, the leeches that are more likely to remain longer in the system are served first over the leeches that in average leave earlier. In the second scheme, the seeds that are more likely to remain in the system are served first over seeds that leave earlier.

4.1 Priority Scheme for Cooperative Leeches

As stated before, the basic idea of the first proposed priority scheme model is to first share the file with the high tolerance leeches in case of penury conditions. It is important to notice

that there can be two cases of penury. The first one occurs when the uploading bandwidth is not enough to serve at the maximum capacity all peers in the system but it is enough to do it for all the cooperative leeches. In this case, the remaining bandwidth is shared among the defeat leeches. The second penury case corresponds to the situation where the uploading bandwidth is not enough to serve the cooperative leeches at maximum capacity. In this case, the defeat leeches are not served at all and the uploading bandwidth is only shared among cooperative leeches. In view of this and based on (4)-(6), the value of u_d can be written as $\min(cx_d, \max(\mu(x+y+1) - cx_c, 0))$.

Let us first focus on the case of abundance. In this condition, the equilibrium values for the number of leeches and seeds in the system are the same as the equations for the two population model in case of abundance, i.e., (7), and (8) with $\beta = c$. Now, for the first case of penury where defeat leeches cannot be served at maximum capacity but there is abundance for the cooperative leeches ($cx > \mu(x+y) > cx_c$), the equilibrium point is given by the following expressions:

$$\bar{x}_{c-P1} = \frac{\lambda_c}{\beta_p \left(1 + \frac{\theta_c}{\beta_p}\right)}, \quad (9)$$

$$\bar{x}_{d-P1} = \frac{\lambda_d + \frac{\lambda_c(c-\beta)}{\beta_p \left(1 + \frac{\theta_c}{\beta_p}\right)}}{\beta \left(1 + \frac{\theta_d}{\beta}\right)}, \quad (10)$$

$$\bar{y}_{P1} = \frac{\beta}{\gamma} (\bar{x}_{c-P1} + \bar{x}_{d-P1}), \quad (11)$$

where

$$\beta_p = \left(\max \left\{ 1/c, (1/\mu - 1/\gamma) \left(\frac{\bar{x}_{c-P1}}{\bar{x}_{c-P1} + \bar{x}_{d-P1}} \right) \right\} \right)^{-1}.$$

Finally, for the second penury case where there is only available bandwidth for the cooperative leeches, i.e., ($cx_c > \mu(x+y)$), the equilibrium values are calculated as follows:

$$\bar{x}_{c-P2} = \frac{\lambda_c - \beta \frac{\lambda_d}{\theta_d}}{\beta \left(1 + \frac{\theta_c}{\beta}\right)}, \quad (12)$$

$$\bar{x}_{d-P2} = \frac{\lambda_d}{\theta_d}, \quad (13)$$

$$\bar{y}_{P2} = \frac{\beta}{\gamma} (\bar{x}_{c-P2} + \bar{x}_{d-P2}). \quad (14)$$

4.2 Priority Scheme for Cooperative Seeds

In this second scheme, the peers that are statistically more likely to remain in the system longer once they have downloaded the file are served first over the peers that defeat once they become seeds. For this model, the number of cooperative and defeat seeds have to be considered separately. Therefore, the two population model presented before has to be extended accordingly. It is important to emphasize the difference between this priority scheme and the priority scheme for cooperative leeches. In the later, each peer is studied regarding their behavior as a leech and the leeches that are statistically more likely to rest in the system are classified as cooperative peers. These peers who have a high probability of staying in the system throughout the download procedure are served first in case of penury. For the former, each peer is studied regarding their behavior as seed and the ones that are statistically more likely to remain in the system as seeds are classified as cooperative peers.

These cooperative peers are served first in case of penury. As such, in the priority scheme for cooperative leeches, the peers are classified as having a high download aborting rate θ_d or a low download aborting rate θ_c while all seeds have the same leaving rate γ . On the other hand, for the priority scheme for cooperative seeds, the leeches are considered as having the same download aborting rate θ while the seeds are classified as having a high leaving rate γ_d and low leaving rate γ_c .

First, the two population model with no priorities is presented. The fluid model presented above is extended for the two types of peers as follows:

$$x'_c(t) = \lambda_c - \theta_c x_c(t) - u_c(t), \quad (15)$$

$$x'_d(t) = \lambda_d - \theta_d x_d(t) - u_d(t), \quad (16)$$

$$y'_c(t) = u_c(t) - \gamma_c y_c(t), \quad (17)$$

$$y'_d(t) = u_d(t) - \gamma_d y_d(t). \quad (18)$$

with $u_c(t)$ and $u_d(t)$ as defined in the Section 3.2. The equilibrium values for the number of each class of leeches and seeds in abundance conditions ($cx_c(t) \leq \mu(x_c + x_d + y_c + y_d + 1)$) are calculated as:

$$\bar{x}_c = \frac{\lambda_c}{c + \theta_c}, \quad \bar{x}_d = \frac{\lambda_d}{c + \theta_d}, \quad (19)$$

$$\bar{y}_c = \frac{c}{\gamma_c} \bar{x}_c - 1, \quad \bar{y}_d = \frac{c}{\gamma_d} \bar{x}_d. \quad (20)$$

However, the case of penury ($cx_c(t) > \mu(x_c + x_d + y_c + y_d + 1)$) is much more complicated. Hence the system in this condition is studied using the Markov model and through the discrete event simulator at the peer level. The Markov chain for this priority scheme for cooperative seeds is given by $(L_c(t), L_d(t), S_c(t), S_d(t))$ with the obvious meaning. Since in this model the interest is to analyze the performance of the system in the presence of cooperative and defeat seeds, the leeches are considered to have the same behavior. Specifically, it is considered that $\theta_c = \theta_d = \theta$ and $\gamma_c < \gamma_d$. The initial state is $(0, 0, 1, 0)$. Starting from state (n_c, n_d, m_c, m_d) , the transition rates are: λ_a to state $(n_c + 1, n_d, m_c, m_d)$ and λ_d to state $(n_c, n_d + 1, m_c, m_d)$; $n_c \theta$ to state $(n_c - 1, n_d, m_c, m_d)$ (for $n_c \geq 1$) and $n_d \theta$ to state $(n_c, n_d - 1, m_c, m_d)$ (for $n_d \geq 1$); $(m_c - 1) \gamma_c$ to state $(n_c, n_d, m_c - 1, m_d)$ (for $m_c \geq 2$) and $m_d \gamma_d$ to state $(n_c, n_d, m_c, m_d - 1)$ (for $m_d \geq 2$); $\min\{cn_c, \mu(n_c + n_d + m_c + m_d)n_c / (n_c + n_d)\}$ to state $(n_c - 1, n_d, m_c + 1, m_d)$ (for $n_c \geq 1$) and $\min\{cn_d, \mu(n_c + n_d + m_c + m_d)n_d / (n_c + n_d)\}$ to state $(n_c, n_d - 1, m_c, m_d + 1)$ (for $n_d \geq 1$).

The fluid model with priorities is basically the same as the model with no priorities except for the rates u_c and u_d . These rates are modified in a similar manner as in the priority model for cooperative leeches and are now described by: $u_c = \min(cx_c, \mu(x + y + 1))$ and $u_d = \min(cx_d, \max(\mu(x + y + 1) - cx_c, 0))$. It is worth noting that when the system is in abundance conditions, the equilibrium values for the number of leeches and seeds are also described by (19)-(20). For the case of penury for all peers in the network, the

equilibrium values are described by:

$$\bar{x}_c = \frac{\frac{\lambda_d \gamma_c}{\theta_d} \left(\frac{\theta_d \lambda_c}{\lambda_d} - \frac{\mu \gamma_c + \mu^2}{\gamma_c} \right)}{\gamma_c (\theta_c + \mu) + \mu^2}, \quad \bar{x}_d = \frac{\lambda_d}{\theta_d}, \quad (21)$$

$$\bar{y}_c = \frac{\mu \left(\frac{\lambda_d}{\theta_d} \right)}{\gamma_c} + \frac{\mu}{\gamma_c} \bar{x}_c - 1, \quad \bar{y}_d = 0. \quad (22)$$

The system in penury only for the defeat seeds is studied through the Markov model and the discrete event simulator due to its complexity. Similarly, the Markov chain is only modified in the following transition rates: $\min\{cn_c, \mu(n_c + n_d + m_c + m_d)\}$ to state $(n_c - 1, n_d, m_c + 1, m_d)$ (for $n_c \geq 1$) and $\min\{cn_d, \max(\mu(n_c + n_d + m_c + m_d) - cx_c, 0)\}$ to state $(n_c, n_d - 1, m_a, m_d + 1)$ (for $n_d \geq 1$).

5 Numerical Results

In this section, we present some relevant results regarding the mathematical models and simulation results of the P2P networks studied above. We first present results considering exponentially distributed random variables to model the upload and download rates. Then we compare these results to the model where constant upload and/or download rates are used in the single population model. Also, we consider fixed transfer rates more suited to model a managed P2P system. Following this, the two population model is numerically studied. Finally, the priority models are extensively analyzed and discussed.

5.1 Exponential Distributed Download and Upload Rates

In this subsection, the exponential distributed model for the transfer rates in the simulation at *peer level* and mathematical models are considered. Recall that the simulation under the exponential model is constructed in order to recreate as close as possible the scenario considered by the Markov and Fluid model presented above, i.e., the download and upload rates are modeled as exponentially distributed random variables with mean c and μ respectively.

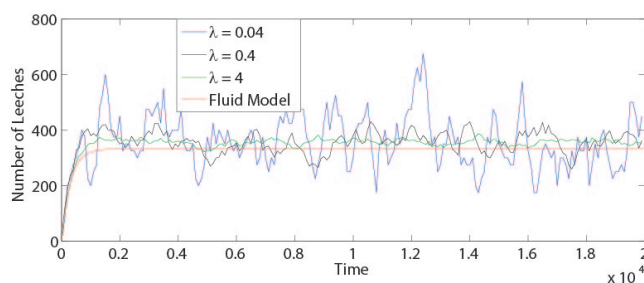
In Fig. 2, the evolution in time of the number of leeches and seeds normalized by the arrival rate

is presented. It can be seen that there is a very good match between the simulation at peer level results compared to the fluid model, specially for high values of λ for both the number of seeds and leeches.

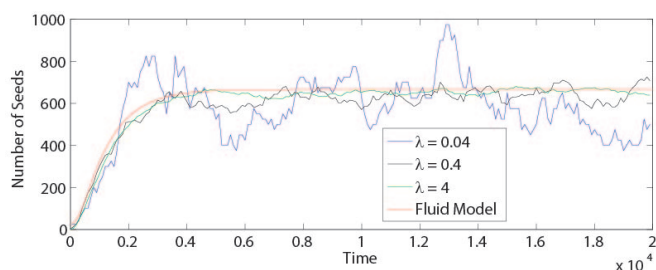
In Fig. 3, the average download (time since the arrival of the peer to the instant that the complete file is downloaded), waiting (time form the arrival of the peer to the moment of the leech departure before downloading the complete file) and life time (time form the arrival of the peer to the moment when the peer either completes de download of the file or leaves the system aborting the download) as leech and the proportion of successful downloads is presented. Various interesting observations can be made from these results.

First of all, it can be seen that the analytical average times derived analytically (under the fluid model), are always the same, independent of the arrival rate. The reason for this can be explained as follows: Any leech can either leave the system either because it abandons the download, with rate θ or because it finishes the download and becomes a seed with rate c (for simplicity consider abundance conditions, i.e., $cx < \mu(x+y)$). Notice that both of these times, i.e. the time that the leech stays in the system before quitting the download and the time that a leech stays in the system before becoming a seed, are independent exponentially distributed random variables. It is well known then, that the minimum of these times can be modeled by a third random variable with rate $c + \theta$. Hence, the average time that a peer stays in the system as leech, either because it downloads the file or because it aborts the download is always $1/(c+\theta)$, which is the value presented in the three graphics for the fluid model.

A second observation is that the simulation results show a slightly higher average times compared to the analytical results. The reason for this is that, in the analytical model, whenever a new peer arrives, it is assumed that it immediately starts to share the file with other peers and at the same time it begins to download the file. On the other hand, the simulation considers that whenever a new peer arrives, it can not upload the file to other peers since it has not yet downloaded the file. And also, if the rest of the peers are already



(a) Leeches



(b) Seeds

Fig. 2. Evolution in time of the number of Leeches and Seeds in the network, Simulation at *peer level* vs Fluid Model

uploading to other peers and there is not an idle peer, it can not start the download of the file. Hence, the download time is slightly higher. And because leeches spend more time in the network before becoming seeds, there is a higher chance of quitting the download, therefore the waiting time is also incremented and consequently the average life time as leech also increases compared to the analytical model. These observations are further reaffirmed by observing the probability that a leech successfully downloads the file in Fig. 3 (d). In the analytical model the probability that a leech becomes a seed is simply $c/(c + \theta)$. In the simulations however, this probability is slightly lower.

5.2 Fixed Download and Upload Rates for Single Population Model

In this subsection, the results when the transfer rates are constant instead of random variables are presented and compared to the exponential model. In Fig. 4, the number of leeches and seeds normalized to λ are presented. For reasons of clarity, only the results for $\lambda = 4$ are shown. The

case where μ and/or c constant are studied. Recall that the exponential model for the transfer rates is not well suited due to its high variability which is not expected to happen in a practical case, specially on a managed P2P network.

It can be seen that by leaving any of the transfer rates constant, the number of leeches is higher than the exponential model and consequently, the number of seeds is smaller. In particular, when both μ and c are fixed, the greatest difference can be appreciated compared to the exponential model. This observation is further confirmed in Fig. 5, where the average number of seeds and leeches is presented. It can be seen that the constant model for both transfer rates also shows the greatest difference for any value of λ . This implies that when both transfer rates are fixed, the average download time is the highest as depicted in Fig. 7(a).

Indeed, it is clear from these results that the exponential model allows a fastest download time and corresponds to an optimistic model for a P2P network. Hence, the average time that a peer stays as leech in the system is also the highest when the

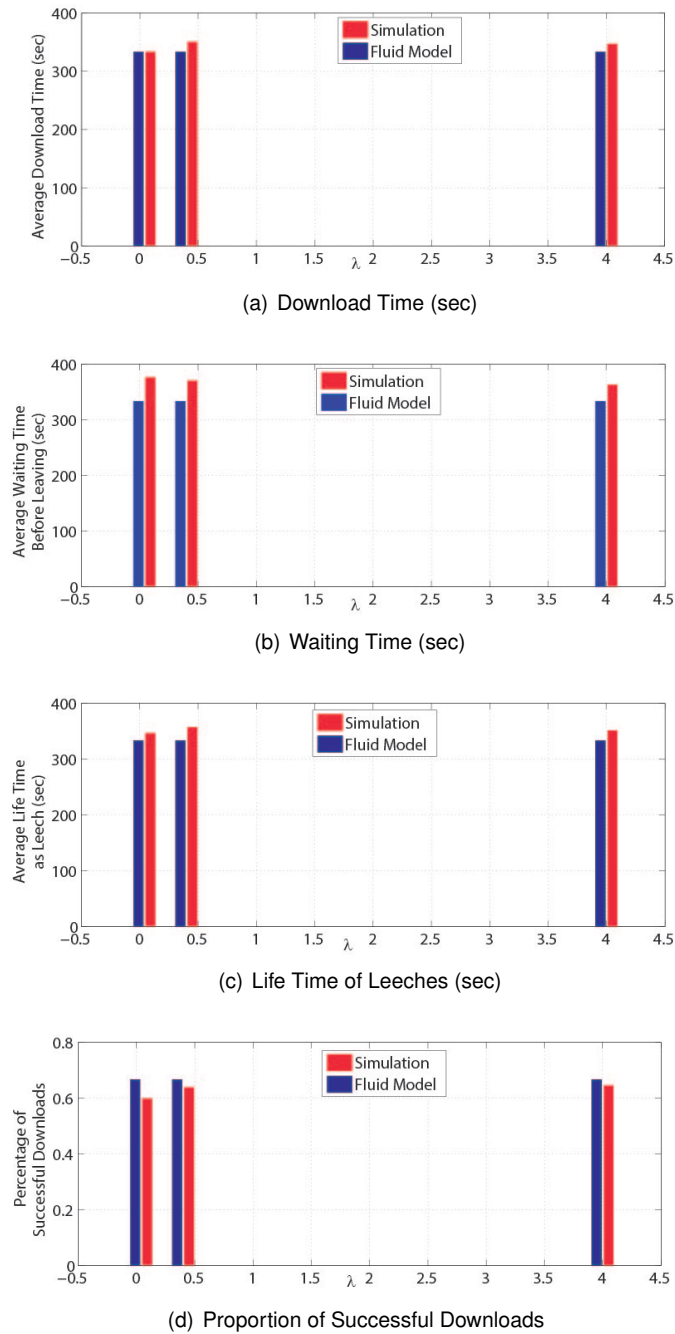


Fig. 3. Average Download, Waiting, Life Times for Leeches (sec) and Proportion of Successful Downloads

transfer rates are fixed as observed in Fig. 7(c) and also, the successful download percentage is the lowest for the same case observed as seen in Fig. 7 (d).

Another interesting observation is that for the case where μ is fixed while c is variable, the variance in terms of the number of leeches and seeds is the highest. This is a good indication that a small variation in the mean value of μ has a big impact on the randomness of the system. However, the overall performance of the system is not considerable different in terms of the download times and probability of successful download.

5.3 Two Population and Priority Models for Exponential Transfer Rates

In this section, the performance of the proposed priority mechanisms are evaluated.

5.3.1 Priority Model for Cooperative Leeches

For these results, different portions of high tolerance leeches are considered, ranging from 1% through 90% of the total leech population. Also, two different system conditions are considered: abundance, where the number of seeds in the system is high enough as to have most of the time all leeches downloading at the maximum capacity. As such, a value of $\gamma = 0.001$, is considered. The second case is penury, where the number of seeds in the system is very low. Hence the uploading bandwidth is very scarce. To obtain this condition, a value of $\gamma = 0.005$ is considered.

The case of abundance can be seen as a case where most peers remain in the system even after the download of the file in order to cooperate with the file download. And the case of penury can be seen as the case where once the file is downloaded, the peers almost immediately leave the system. It is important to keep in mind that seeds' behavior analysis is a different issue than the one investigated in this work. Recall that we are focusing on the behavior of the leeches while they are downloading the file, i.e., if they have a high probability of aborting the download or not. Nevertheless, this priority scheme can also be implemented focusing on the behavior of the seeds, which we plan to study in a future work.

Another important remark about the results presented in this section is that, as stated in [2], the fluid model is more accurate when the arrival rate λ is high. Hence, we use the fluid model for $\lambda = 4$, while the Markov model is used to obtain results for $\lambda = 0.04, 0.4$; in order to verify the analytical results, the value of $\lambda = 4$ is also considered.

In Fig. 8, the normalized equilibrium values for the number of leeches and peers under the proposed priority scheme is presented using the fluid model when $\lambda = 4$ and for different proportions p of high tolerance leeches in abundance conditions.

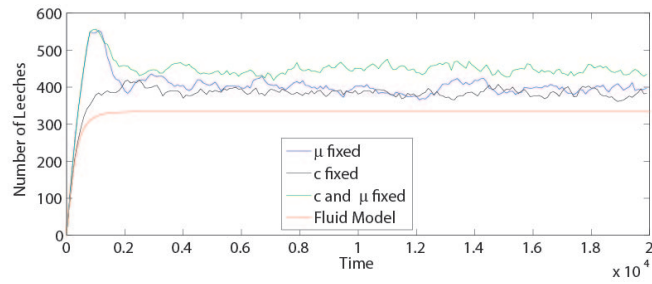
The normalization is made with respect to λ . In Fig. 9 we present the same results but using the Markov model and for $\lambda = 0.04, 0.4$ and 4. First, it is important to remark that both models present a very close match, validating somehow each other. Also, under this condition of abundance, the values for the normalized number of leeches and seeds is exactly the same for the system with and without priorities as described in previous section.

Indeed, the priority scheme has no impact on the system when the uploading bandwidth is high enough as to allow all leeches to download at the maximum capacity c .

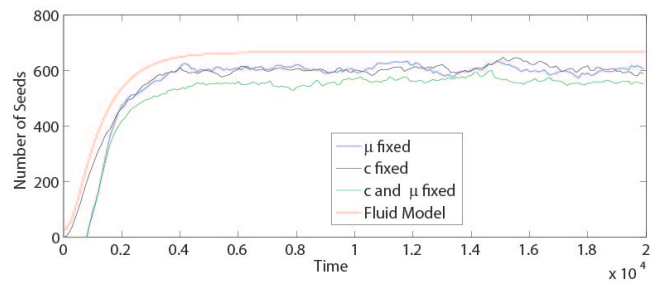
In Fig. 10, the probability that a leech successfully downloads the file is presented. Again, there is no difference between the conventional system and the system with the priority scheme enabled. It is important to notice that as the number of high tolerance leeches increases, the probability of successful download also increases.

This is because high tolerance leeches remain longer in the system and have a high probability of remaining throughout the whole downloading process. Conversely, when the fraction of high tolerance leeches is low (for example $p = 0.01$), there is a high probability that the downloading leeches abort the download and leave the system.

In penury conditions, the priority mechanism has a visible impact on the system since, in that case, high tolerance leeches are served first. For instance, Fig. 11 shows the normalized equilibrium value of the number of leeches and seeds for $\lambda = 4$ derived from the fluid model. It can be seen that when the priority mechanism is enabled,

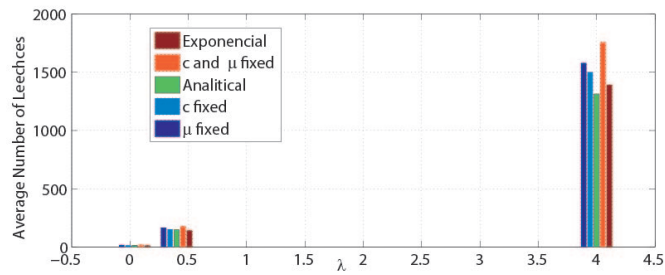


(a) Number of Leeches

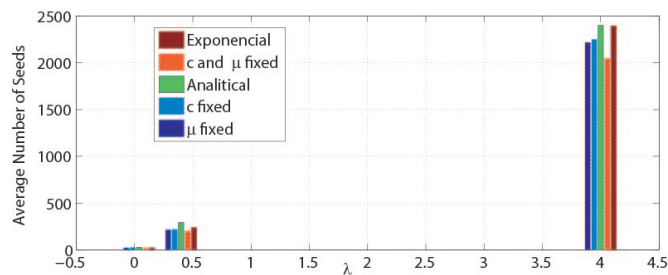


(b) Number of Seeds

Fig. 4. Evolution in time of the number of Leeches and Seeds in the network with $\lambda = 4$

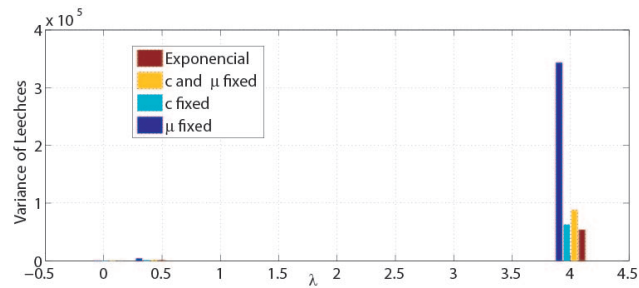


(a) Average Number of Leeches

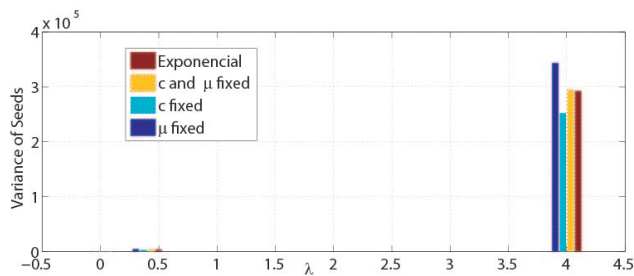


(b) Average Number of Seeds

Fig. 5. Average Number of Leeches and Seeds for c and/or μ Constant



(a) Variance of the Number of Leeches



(b) Variance of the Number of Seeds

Fig. 6. Variance of the Number of Leeches and Seeds for c and/or μ Constant

the normalized number of high tolerance leeches is higher than for the conventional system. More importantly however is the fact that the normalized number of seeds when the priority mechanism is enabled is higher than for the conventional case.

This effect is more visible when there is a high portion of high tolerance leeches in the system, since in this case there is more cooperation among peers. Therefore, there are more leeches that download the file and become seeds. This is “paid” by the low tolerance leeches that are not being served and leave the system before having a chance of downloading the file.

This observation is reinforced by observing the results in Fig. 13 (a) which shows the probability of successfully downloading the file. It can be seen that for different values of λ this probability is always higher for the high tolerance leeches when the priority scheme is enabled. In fact, for high values of λ and high percentage of high tolerance leeches in the system, whenever a high tolerance leech initiates the download of the file, it will successfully download it. In other words,

for these leeches, the probability to successfully download the file is (very close to) one.

Conversely, the probability that a low tolerance leech downloads a file is almost zero. Recall that low tolerance leeches are not rewarded, but notice that not rewarding the low tolerance leeches is different from being punished. The difference is that the peers are not wasting resources in punishing these leeches, they are just *ignoring* them whenever the system is in penury conditions. Since the low tolerance leeches are not likely to remain in the system throughout the complete download process, the global resources are better used by serving first the more tolerant users. Also, the less tolerant leeches do not get served unless there are enough resources in the system, and they are more likely to abort the download reducing significantly the successful download probability.

Notice that the results from the Fluid Model in Fig. 11 are validated by the results from the Markov model presented in Fig. 12. Observe that there is a better fit between these two models when λ is high, as already mentioned in [2].

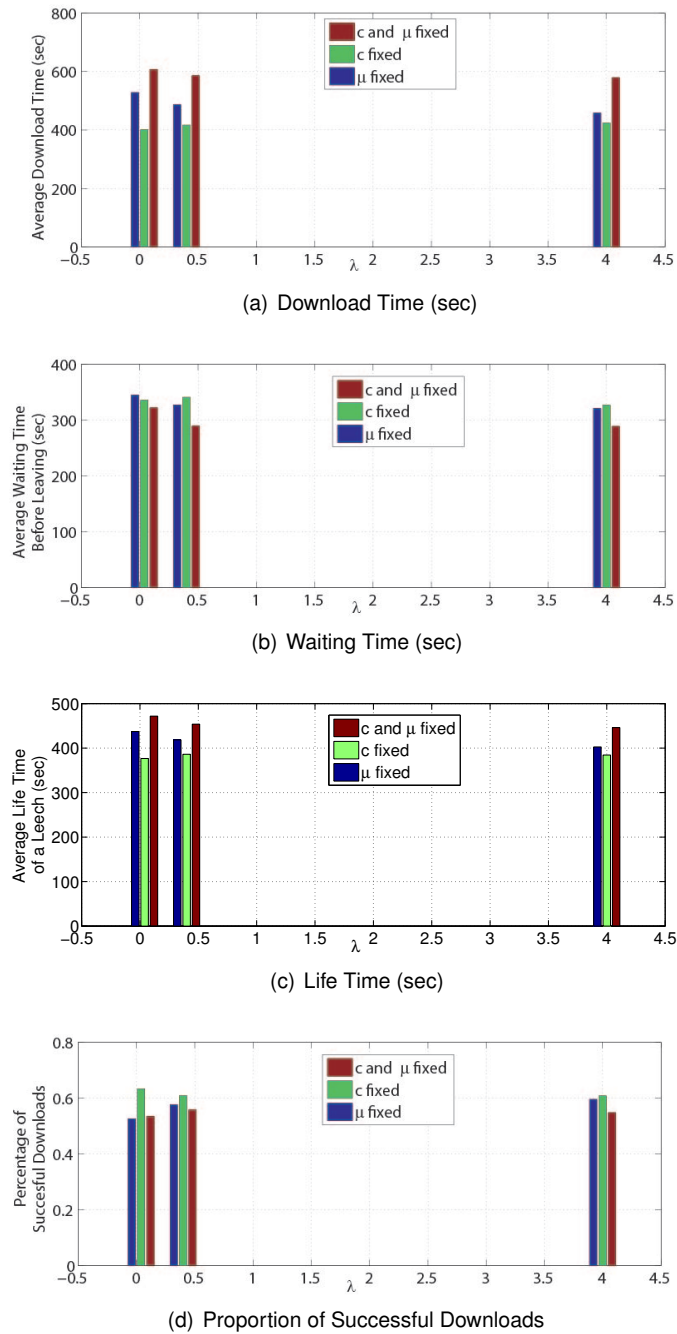
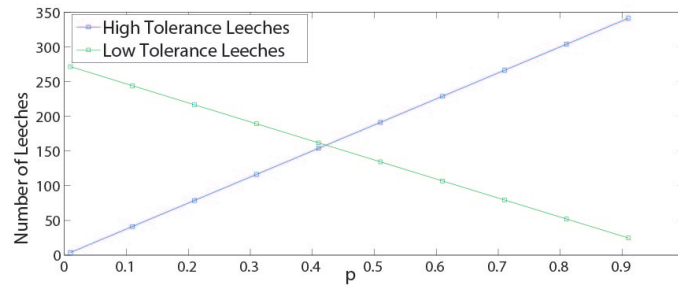
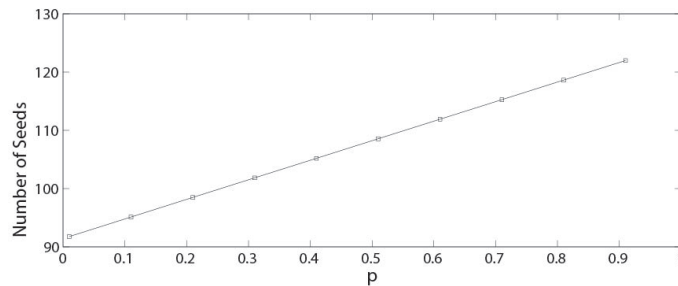


Fig. 7. Average Download, Waiting, Life Time of Leeches and Proportion of Successful Downloads

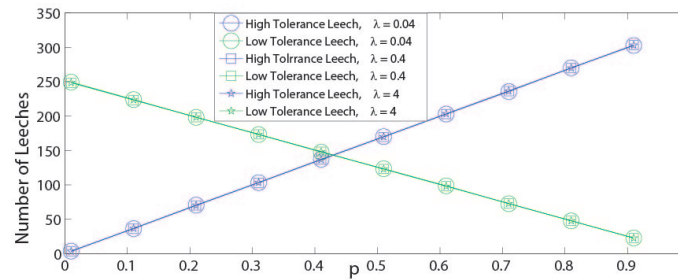


(a) Leeches

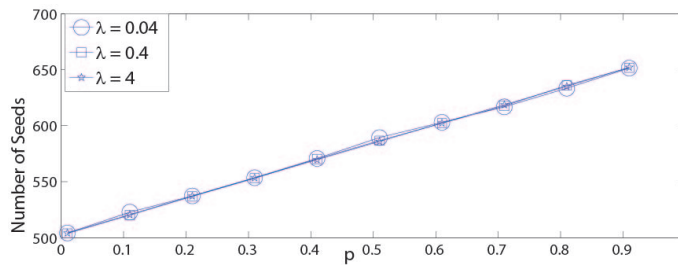


(b) Seeds

Fig. 8. Normalized Equilibrium Number of Leeches and Seeds for Different Portions of High Tolerance Leeches, Fluid Model, Abundance



(a) Leeches



(b) Seeds

Fig. 9. Normalized Equilibrium Number of Leeches and Seeds for Different Portions of High Tolerance Leeches, Markov Model, Abundance

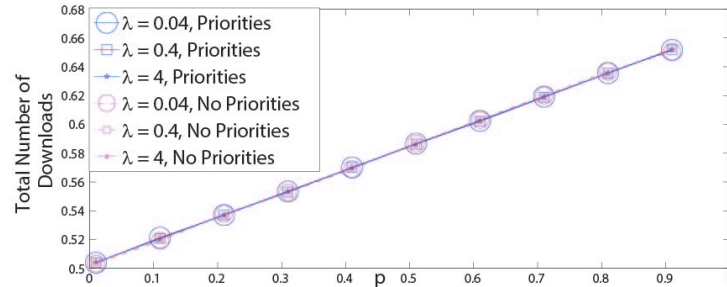
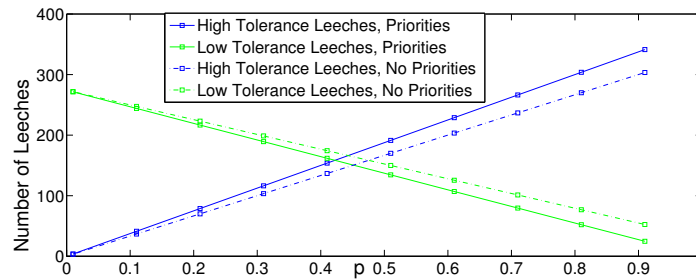
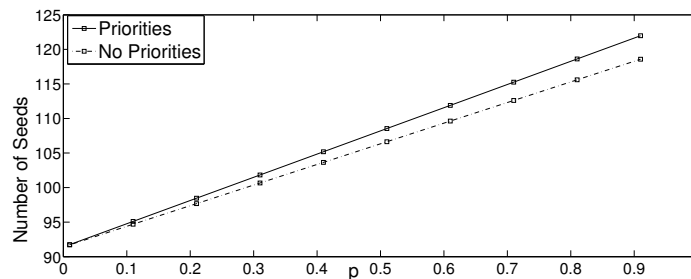


Fig. 10. Probability of Total Successful Downloading, Abundance

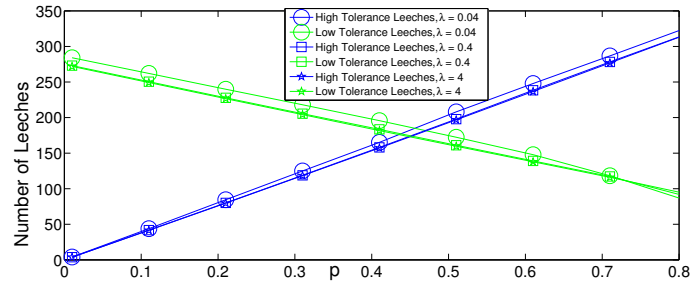


(a) Leeches

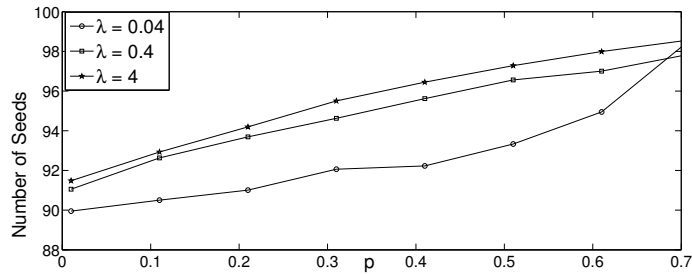


(b) Seeds

Fig. 11. Normalized Equilibrium Number of Leeches and Seeds for Different Portions of High Tolerance Leeches, Fluid Model, Penury

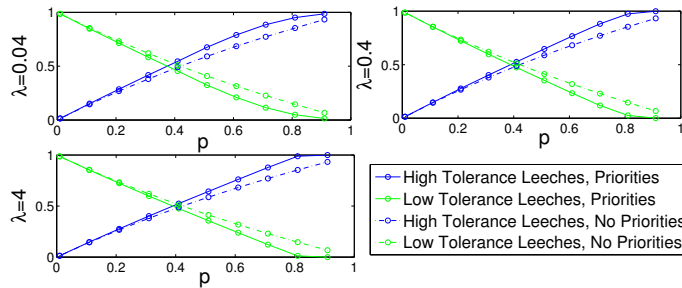


(a) Leeches

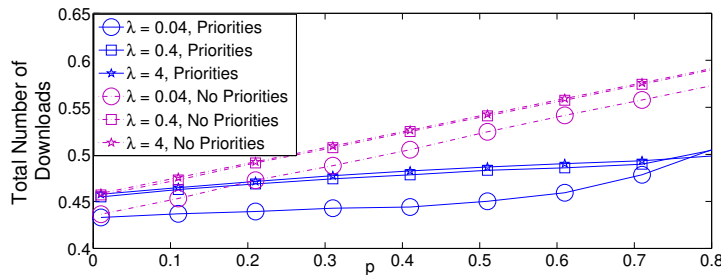


(b) Seeds

Fig. 12. Normalized Equilibrium Number of Leeches and Seeds for Different Portions of High Tolerance Leeches, Markov Model, Penury



(a) Probability of Successful Downloading for Each Class of Leeches, Penury



(b) Probability of Total Successful Downloading, Penury

Fig. 13. Successful download in penury

Finally, the total successful download probability is presented in Fig. 13 (b). In this case, the percentage of leeches (both high and low tolerance) is lower when the priority scheme is enabled compared to the conventional system. The reason behind this is that, as already stated, the high tolerance leeches are rewarded by serving them before the low tolerance ones.

Hence, these low tolerance leeches, which already have a low latency tolerance, are more prone to leave the system. That is, for the conventional system with the priority scheme disabled, the probability that some low tolerance leeches download the file is not zero because the peers cooperate with any leech independently of its tolerance to the download delay. As such, the probability that a leech in general downloads the file is higher when no priority scheme is being used.

5.3.2 Priority Model for Cooperative Seeds

In this section, the case where the cooperative seeds are served first is discussed in detail. For these experiments, the system parameters are: $c = 0.002$, $\mu = 0.0012$, $\theta = 0.001$, $\lambda_c = p\lambda$, $\lambda_d = (1 - p)\lambda$, $\gamma = 0.001, 0.005$, $\gamma_c = \gamma$, and $\gamma_d = 5\gamma$. We highlight the main difference between these two models. In the Markov model, the cooperative peers use the available bandwidth among each other while the defeat peers can only use the remaining bandwidth. However, for the simulation model, not all peers can cooperate among each other. Recall that arriving peers look for available peers until they achieve the maximum download rate or no active peer can download him. Hence the case where no available peers are found can occur. For the priority schemes, when a cooperative peer arrives, it looks for peers to download the file and if it does not find one, it looks for a peer that is serving a defeat peer. If there is one defeat peer being served the downloading peer stops serving the defeat leech and starts to serve the cooperative leech. This cooperative peer continues this procedure until it achieves the maximum downloading rate or no more peers are available. This issue provides

important differences among the results of the two models.

Let us first discuss the Markov model. In Fig. 14 the normalized (on λ) number of peers for the priority scheme that favors the cooperative seeds when the system is stable for $\gamma = 0.001$ is presented while in Fig. 16 the same results are presented for $\gamma = 0.005$. As such, Fig. 16 corresponds to a harsher environment compared to the case presented in Fig. 14, since seeds remain shorter times in the system. In Fig. 15, the total download probability is depicted for $\gamma = 0.001$. It is worth mentioning that for low concentrations of cooperative seeds, the probability of downloading the file is slightly lower than for the case of high concentration of cooperative peers. In other words, this probability increases as the concentration of cooperative peers increases until a point (close to 20%) where it stabilizes. The reason for this is that, when $\gamma = 0.001$ the system has enough seeds (even if many of them are non cooperative) to serve almost any leech arriving to the system. Hence, most of the peers that do not abort the download become seeds. As such, any increment on p after 0.2 has no visible effect on the probability of downloading the file. On the other hand, observe Fig. 17, where γ is much higher (i.e., seeds remain in the system for shorter times), there are not enough resources in the system to serve arriving leeches. Therefore, any increase on the concentration of cooperative seeds, increases the probability of successful download. In this sense, it is fair to say that for $\gamma = 0.01$ the system is in abundance for $p \geq 0.2$, while for $\gamma = 0.005$, the system is in penury for any value of p . Comparing the performance of the system under the priority scheme that benefits the cooperative seeds with the scheme that favors the cooperative leeches, observe Figs. 10, 13 (b), Fig. 15, and Fig. 17. It is clear that the priority scheme that favors cooperative seeds has a better performance in terms of total download probability than the priority scheme that favors the cooperative leeches. Also note that, when the cooperative seeds are prioritized, in case of penury, the probability of successful download is higher than the case where no priorities are enabled. Recall that the priority scheme that favors

cooperative leeches, in the same conditions of penury, the probability of successful download is higher when no priorities are enabled.

We now consider the simulation's results. The performance of the system under the two priority schemes is presented. For $\gamma = 0.001$, Fig. 18 presents the average number of peers in the system for the priority scheme for cooperative leeches while Fig. 19 presents the results for the priority scheme for cooperative seeds. It is clear that for the first priority scheme, there is only one class of seeds. Hence, only the total number of seeds is shown.

On the other hand, for the second priority scheme, the two classes of seeds is shown. From these results it is clear that as the number of cooperative peers increases, also the total number of seeds increases. This is because the peers rest in the system longer times either as leeches, completing a higher number of downloads, or as seeds, remaining longer times after the download.

In Fig. 20, the average download time for the two priority schemes and the system with no priorities is presented. For reasons of clarity, only the results for $\lambda = 0.4$ and $\lambda = 4$ are shown. In Fig. 20 (a) the performance of the system for the cooperative peers is presented. It can be seen that the lowest download times corresponds to the scheme where the cooperative seeds are benefited. As expected, when the peers that are statistically more likely to remain longer in the system after the download are served first, there is a higher probability of finding a peer with the complete file and hence, the file is downloaded faster.

For the priority scheme that favors the cooperative leeches, the download time is higher because it only benefits the peers that stay longer during the download procedure. However, these peers may quit the system almost immediately after downloading the file. As such, it is more profitable to benefit the peers that stay longer sharing their resources once they complete their download than favoring those who are more likely to stay throughout the download but may act selfishly afterwards.

Also, when no priorities are enabled in the system, the average download time is the highest. The highest download time for the defeat peers

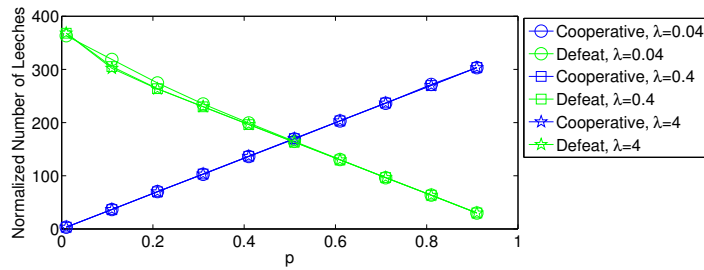
corresponds to the priority scheme that favors the cooperative seeds. Again, this is due to the fact that defeat peers are only served after the cooperative peers are served. Hence it takes more time for them to download the file. One important issue is the total download time in the system.

It can be seen that the priority scheme for the cooperative leeches achieves the lowest download times for all peers in the system while the priority scheme for cooperative seeds achieves a low download time only after a certain number of cooperative seeds are in the system.

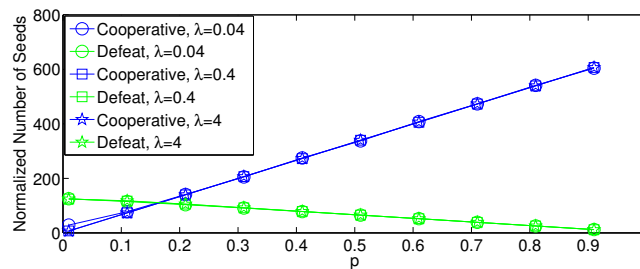
The rationale behind this can be explained as follows: when the cooperative seeds are favored and the number of cooperative seeds is low, the system serves first these few peers which does not have a great impact on the system since most of the seeds quit the system very fast after their download.

Hence, the download time for the rest of the peers (which are defeat seeds) is high. As the proportion of cooperative peers increases, the probability of having cooperative seeds in the system increases, reducing also the download time. For the case when the cooperative leeches are favored, when the number of cooperative peers are in the system the download time is low since the model considers that all seeds remain in average the same amount of time ($1/\gamma$), therefore, the number of seeds is high compared to the second priority scheme (where most seeds remain only $1/\gamma_d$ which is a much shorter time).

In Fig. 21, the proportion of successful downloads for the two priority schemes and the total number of downloads is presented. In the case of the cooperative peers, the scheme that achieves the best results is the one that favors the cooperative leeches, as it is shown in Fig. 21 (a). Conversely, for the defeat peers the priority scheme that benefits the cooperative peers achieves the highest proportion of successful downloads. Also, this scheme obtains the highest total proportion of successful downloads for all peers in the system. Indeed, as the system serves first the peers that as seeds stay longer in the system, all peers benefit from having a higher probability of finding a peer with the complete file



(a) Leeches



(b) Seeds

Fig. 14. Number of Peers for Different Portions of Cooperative Peers, Priorities for Seeds, $\gamma = 0.001$, Markov Model

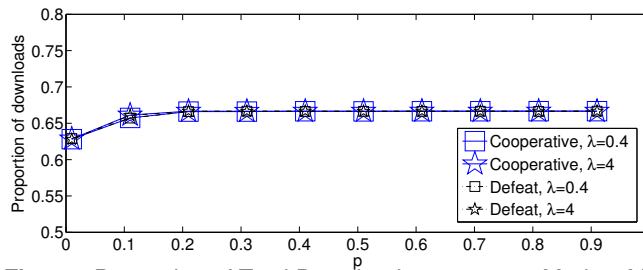


Fig. 15. Proportion of Total Downloads, $\gamma = 0.001$, Markov Model

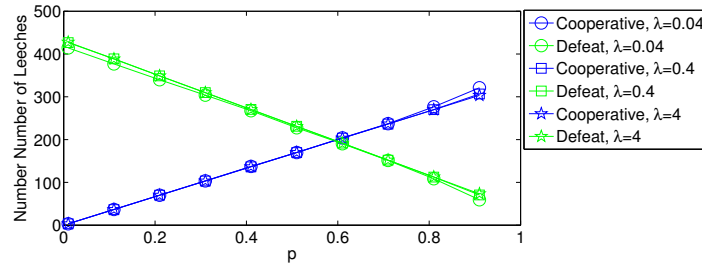
willing to cooperate. As such, the proportion of total successful downloads is higher.

On the other hand, for the cooperative peers, the scheme that benefits the seeds that remain longer has the lowest proportion of downloads specially for low concentration of cooperative peers. This is because there is a high number of non cooperative peers which leave the system almost immediately after downloading the file. In the same conditions, the priority scheme that benefits the cooperative leeches, considers that all seeds have the same rate γ , which implies a higher number of seeds. This in turn increases the number of downloads since the leeches benefit from a high number

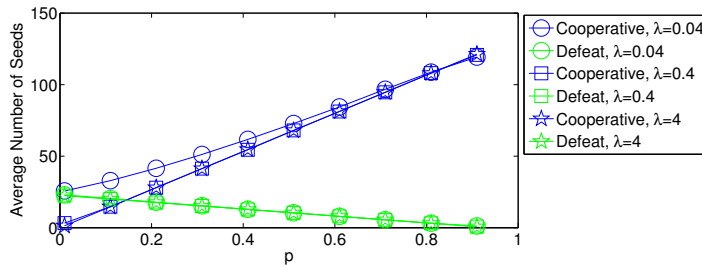
of seeds irrespectively of the concentration of cooperative leeches.

5.4 Two Population and Priority Models with Constant Transfer Rates

In this section, the effect of the constant transfer rates are discussed in detail. The cases where the download and/or the upload rates are constant are considered. For these experiments the system parameters are as follows: $\lambda = 0.4$, $\theta = 0.01$, $c = 0.002$, $\mu = 0.00125$, $\gamma = 0.01$ for abundance conditions and $\gamma = 0.05$ for penury conditions. First, we focus on the average download time.



(a) Leeches



(b) Seeds

Fig. 16. Number of Peers for Different Portions of Cooperative Peers, Priorities for Seeds, $\gamma = 0.005$, Markov Model

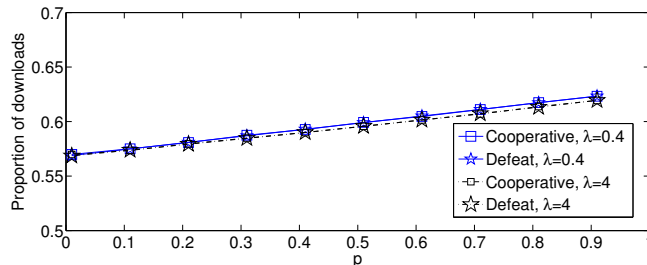


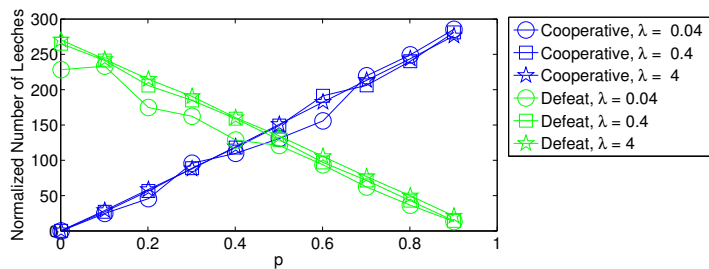
Fig. 17. Proportion of Total Downloads, $\gamma = 0.005$, Markov Model

It can be seen in Fig. 22 that in the system where c and/or μ are constant the average download time increases compared to the exponential model under abundance conditions. This implies that the high variability on the transfer rates implicit on the exponential model corresponds to a *optimistic* system in the sense that the system is benefited by allowing both very high and very low values for these transfer rates, which is not expected on a managed P2P network.

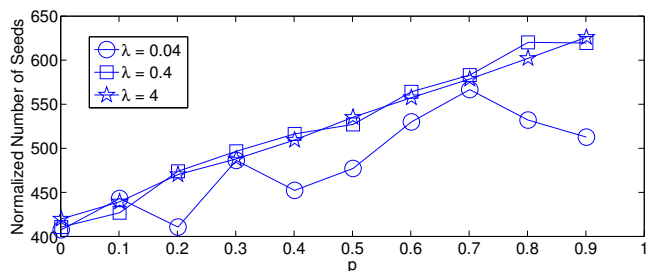
Also, from these results, it can be seen that by considering only c fixed and μ exponentially distributed, the system behaves in the same manner as the only exponential model, i.e., the

system with the priority scheme for cooperative leeches and the system with no priorities start from a low value of download time and increases with p while the system under the priority scheme for cooperative seeds starts from a high value of download time and decreases for p .

Conversely, when μ or both transfer rates are constant, the behavior of the system is the opposite. As such, it is clear that the system is very sensitive on the model used for the parameter μ . The same observations can be drawn for the system under penury conditions as it is shown in Fig. 23.

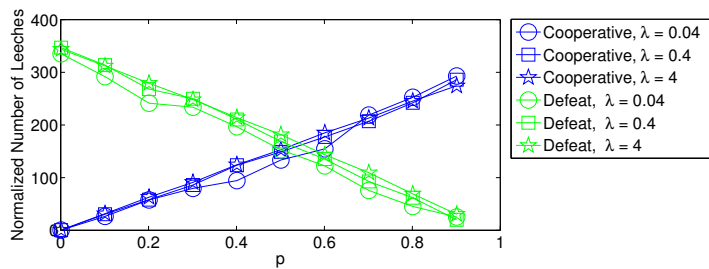


(a) Leeches

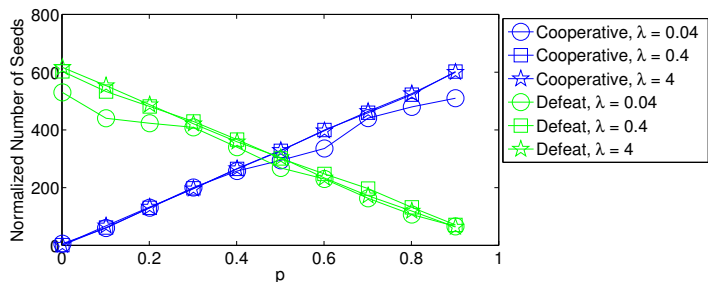


(b) Seeds

Fig. 18. Number of Peers for the Priority for Cooperative Leeches Scheme, Simulation, $\gamma = 0.001$

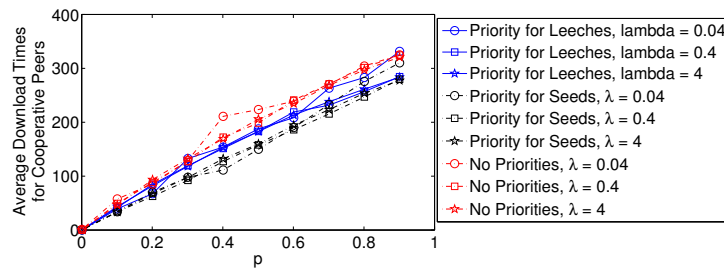


(a) Leeches

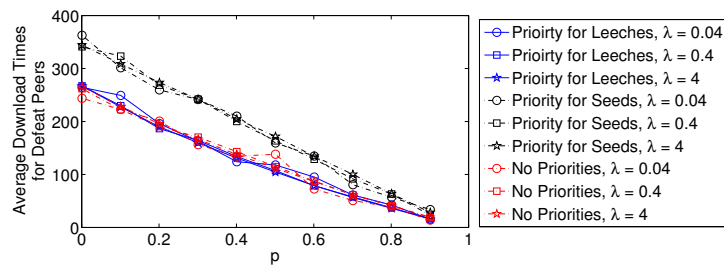


(b) Seeds

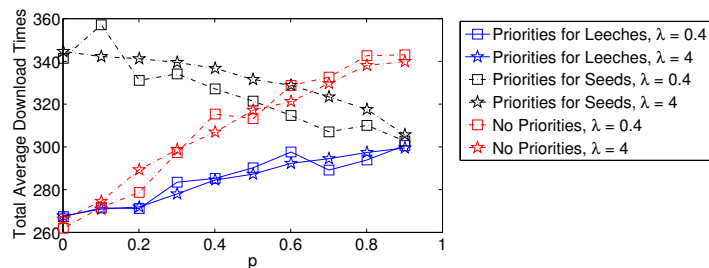
Fig. 19. Number of Peers for the Priority for Cooperative Seeds Schemes, Simulation, $\gamma = 0.001$



(a) Cooperative Peers



(b) Defeat Peers



(c) Total

Fig. 20. Average Download Time for Priority Schemes, Simulation, $\gamma = 0.001$

Therefore, the impact of the model of the transfer rates is independent on the system's conditions. It is worth noting that the priority schemes are no longer effective when μ and/or both transfer rates are constant, since for high concentrations of cooperative peers, the system with no priorities has the best performance.

Similarly, for the probability of successful download, a similar effect is presented as shown in Figs. 24 and 25. First, this probability is lower for the all exponential case. Second, the case where only c is constant provides a similar behavior than the exponential model where the system with no priorities has the worst performance for any value of p while for the models where μ and/or both

transfer rates are constant for high concentration of cooperative peers, the system with no priorities obtains the best results, i.e., the behavior of the system drastically changes.

Again, we conclude that the parameter μ is highly sensitive to the selected model. It is worth noting that, for any model used, and for low concentrations of cooperative peers, the priority scheme that favors the cooperative seeds, obtain a much higher probability of successful download. These are encouraging results since this priority scheme can be used for both managed or unmanaged P2P networks in order to increase the successful download probability in particular for

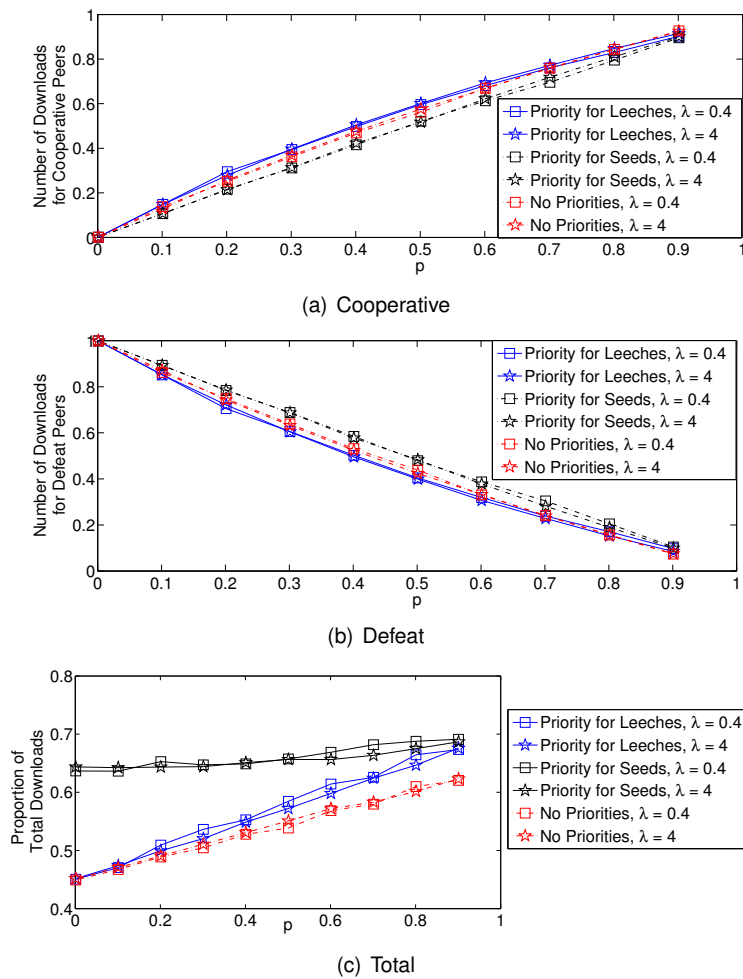


Fig. 21. Proportion of Downloads for the Priority Schemes, Simulation, $\gamma = 0.001$

a system where a high portion of seed behave in a selfish manner.

6 Conclusions

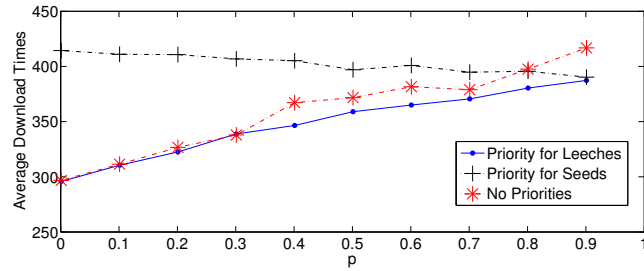
This work has focused on studying the benefits of serving first the peers that have a high download delay tolerance over peers that have very low tolerance. The resulting P2P networks have been modeled, simulated, and studied. The model developed describes a BitTorrent-like network. It calculates the number of leeches and seeds in the system in equilibrium and the

probability of successful download for different system conditions.

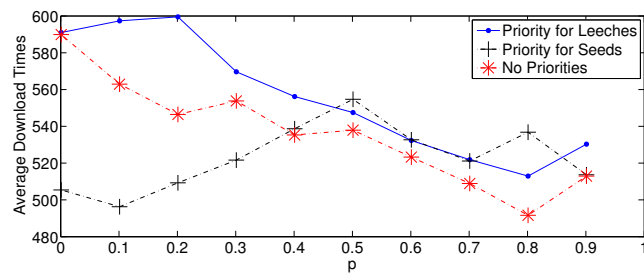
The results are obtained using both a simple fluid model and a Markov chain model. It has been shown that the priority schemes are well suited when the uploading bandwidth in the system is scarce.

As such, it prevents for resources wastage by attending first the leeches that are more likely to become seeds as well as the case where peers that are more likely to remain longer in the system once they become seeds are served first.

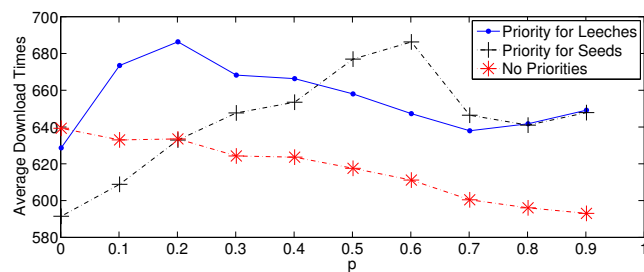
Consequently, these cooperative peers can share their resources with the rest of peers. It is



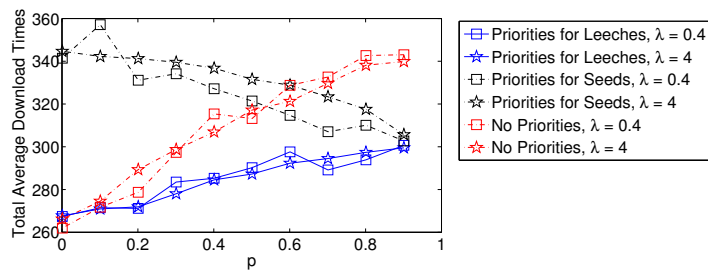
(a) Constant c



(b) Constant μ

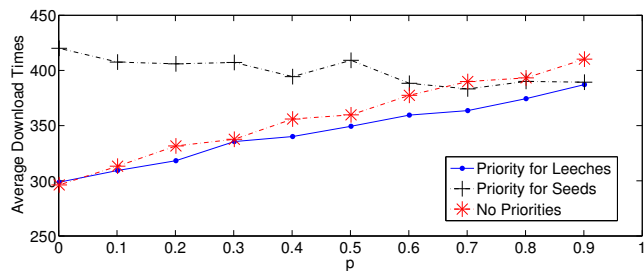


(c) Constant c and μ

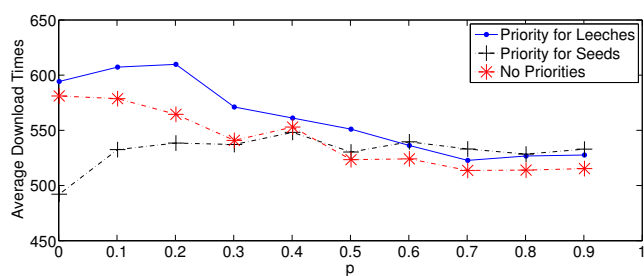


(d) Exponential model

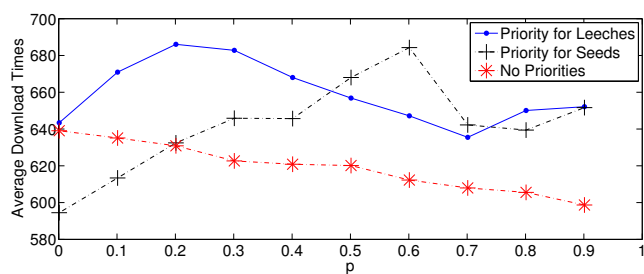
Fig. 22. Average Download Time for Priority Schemes, Abundance



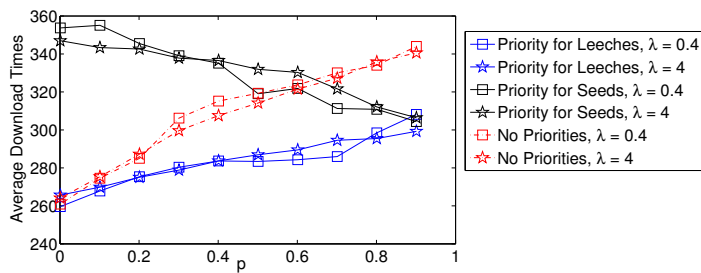
(a) Constant c



(b) Constant μ



(c) Constant c and μ



(d) Exponential Model

Fig. 23. Average Download Time for Priority Schemes, Penury

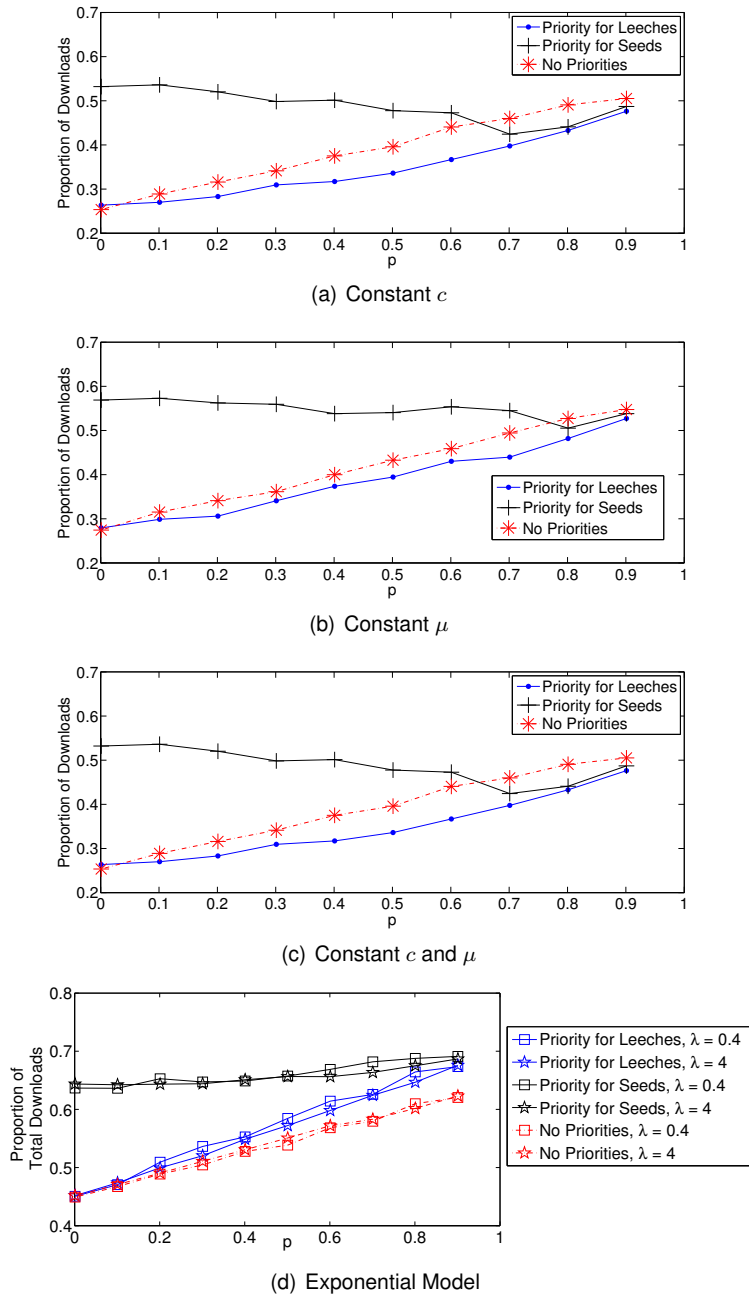
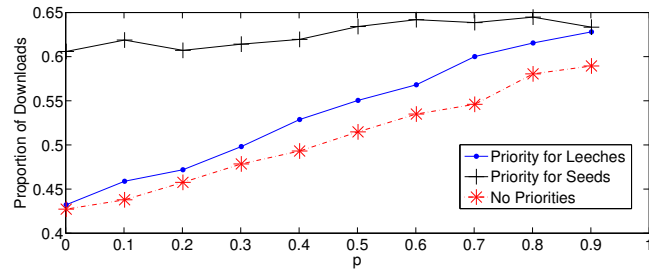
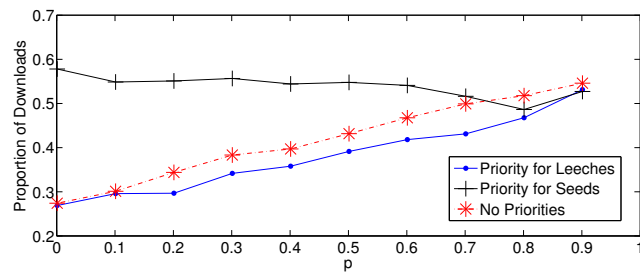


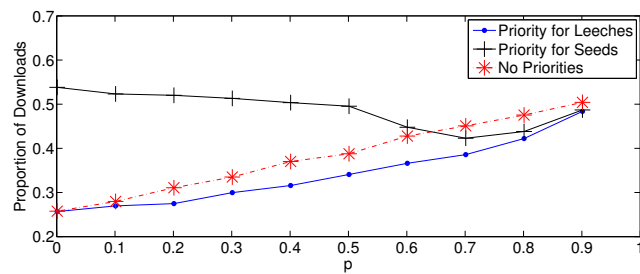
Fig. 24. Proportion of Total Downloads for Priority Schemes, Abundance



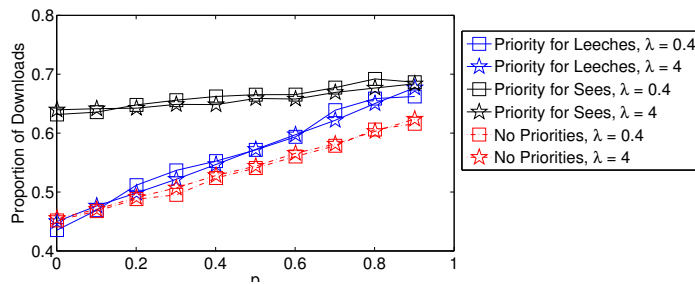
(a) Constant c



(b) Constant μ



(c) Constant c and μ



(d) Exponential Model

Fig. 25. Proportion of Total Downloads for Priority Schemes, Penury

important to mention that the BitTorrent application is considered in this work for reasons of simplicity.

However, the proposed priority mechanism can be applied to other P2P applications. Also, since the objective of this work is to study the potential benefits of rewarding cooperative leeches in P2P networks, it can be used as a guideline for future practical implementations.

This work also studies different models for the transfer rates, μ and c . Since the exponential model has a big variance which is not well suited for a real system, the fixed model is considered as alternatives which corresponds to a good fit for managed P2P networks. From the results presented in this work, it can be seen that the exponential model corresponds to an optimistic model in the sense that the average download times are lower and probability of success are higher than for the other models.

Also, it can be seen that the parameter λ has an important impact on the performance of the system when the constant model is used. Another important observation is that the priority scheme that benefits the cooperative seeds provides a much higher successful download probability for both exponential and constant models for a system where most seeds behave in a selfish manner.

In further work, we will exploit more in depth these models by proposing different distributions for the transfer rates and we will extend them to analyze other factors of global performance not considered here, such as the abort download rate θ and seed's departure rate γ . These models are of special interest in managed networks where the system managers have more control on the behavior of the peers.

References

1. **Rand, D.G., Dreber, A., Ellingsen, T., Fudenberg, D., & Nowak, M.A. (2009).** Positive interactions promote public cooperation. *Science Magazine*, Vol. 325, pp. 1272–1275.
2. **Qiu, D. & Srikant, R. (2004).** Modeling and performance analysis of BitTorrent-Like Peer-to-Peer networks. *Proceedings ACM SIGCOMM*, Vol. 3.
3. **Ge, Z., Figueiredo, D.R., Jaiswal, S., Kurose, J., & Towsley, D. (2003).** Modeling Peer-Peer file sharing systems. *Proceedings IEEE INFOCOM*, Vol. 3, No. 3, pp. 2188–2198.
4. **Petrovic, S. & Brown, P. (2009).** Large scale analysis of the eDonkey P2P file sharing system, *Proceedings IEEE INFOCOM*, pp. 2746–2750.
5. **Kesidis, G., Jin, Y., Mortazavi, B., & Konstantopoulos, T. (2006)**An epidemiological model for file-sharing with BitTorrent-like incentives: The case of a fixed peer population. *Proceedings IEEE Globecom*, Vol. 1, pp. 1–5.
6. **Efthymiopoulos, N., Tompros, S.L., & et al. (2011).** Enabling live video streaming services realization in telecommunication networks using P2P technology. *International Journal of Communication Systems* Vol. 24, No.10, pp. 1354–1374. DOI: 10.1002/dac.1250.
7. **Sing-Ling, C. (2011).** Enhancing P2P live streaming performance by balancing description distribution and available forwarding bandwidth in P2P streaming network. *International Journal of Communication Systems*, Vol. 24, No. 5, pp. 568–585. DOI: 10.1002/dac.1173.
8. **Ying-Xun, L., Chin-Feng, L., & et al. (2011).** A personalized mobile IPTV system with seamless video reconstruction algorithm in cloud networks. *International Journal of Communication Systems*, Vol. 24, No. 10, pp. 1375–1387. DOI: 10.1002/dac.1254.
9. **Lloret, J., García M., Atenas, M., & Canovas, A. (2011).** A QoE management system to improve the IPTV network. *International Journal of Communication Systems*, Vol. 24 No. 1, pp. 118–138. DOI: 10.1002/dac.1145.
10. **Jing-cheng, O., Yue-bin, W., & Xu-huai, H. (2009).** An incentive mechanism using game theory for P2P networks. *Proceeding IEEE International Conference on Networks Security, Wireless Communications and Trusted Computing*, Vol. 2, No.25-26, pp. 715–718.
11. **Bocek, T., Shann, M., Hausheer, D. & Stiller, B. (2008).** Game theoretical analysis of incentives for large-scale, fully decentralized collaboration networks. *Proc. IEEE International Symposium on Parallel and Distributed Processing*, pp. 1–8.
12. **Liao, W.C., Papadopoulos, F., & Psounis, K. (2007).** Performance analysis of BitTorrent-Like systems with heterogeneous users. *Elsevier Performance Evaluation*, Vol. 64, No. 9-12, pp. 876–891.

13. **De Veciana, G. & Yang, X. (2003).** Fairness, incentives and performance in peer-to-peer networks. *Proc. Forty-First Annual Allerton Conference on Communication, Control and Computing*, pp. 749–758.
14. **Ramachandran, K.K. & Sikdar, B. (2010).** A queuing model for evaluating the transfer latency of peer-to-peer systems. *Parallel and Distributed Systems, IEEE Transactions*, Vol. 21, No. 3, pp. 367–378. DOI: 10.1109/TPDS.2009.69.
15. **Weijie, W. & Lui, J. (2012).** Exploring the optimal replication strategy in p2p-vod systems: Characterization and evaluation. *Parallel and Distributed Systems, IEEE Transactions*, Vol. 23, No. 8, pp. 1492–1503. DOI: 10.1109/INFCOM.2011.5934900.
16. **Mizanian, K., Vasef, M., & Analoui, M. (2010).** Bandwidth modeling and estimation in peer to peer networks. *International Journal of Computer Networks & Communications*, Vol. 2, No. 3, pp. 65–83. DOI: 10.5121/ijcnc.2010.2306.
17. **Kotevski, Z. & Pece, M. (2011).** A Modeling Framework for Performance Analysis of P2P Live Video Streaming Systems. In: *ICT Innovations*, Springer Berlin Heidelberg, pp. 215–225.

Article received on 04/07/2017; accepted on 29/05/2018.
Corresponding author is Mario E. Rivero-Angeles.