

# Performance Analysis in Peer-to-Peer Networks with Collaborative Initial States

Gerardo Hernández-Oregón, Mario E. Rivero-Angeles\*, Juan C. Chimal-Eguía,  
Jorge E. Coyac-Torres

Instituto Politécnico Nacional,  
Centro de Investigación en Computación,  
Mexico

{ghernandezo1101, jcoyact1900}@alumno.ipn.mx, {mriveroa, jchimale}@ipn.mx

**Abstract.** Currently, the trends of the new generations of telecommunications (6G and beyond) point towards a non-centralized process, however, for these architectures where no central entity regulates the transmission and information sharing, collaboration among nodes becomes a major issue to provide an adequate service. To this end, in this paper, we study Peer-to-Peer (P2P) systems to obtain models that reproduce these technologies' operation and the collaboration mechanism. The behavior of these systems is modeled through a Continuous Time Markov Chain (CTMC), and later we develop a Discrete Event Simulation (DES) to validate the analytical results. For solving these processes, M/M/s type queues were used in order to analyze the impact of collaboration levels on the system performance.

**Keywords.** Peer-to-peer (P2P) systems, continuous time Markov chain (CTMC), collaboration mechanism, discrete event simulation (DES), collaborative threshold ( $\zeta$ ).

## 1 Introduction

As a result of the creation of the first computer network, new topologies and data structures began to be implemented, in particular, centralized architectures where a server acts as the entity in charge of distributing information through the network. However, these architectures suffer from bottlenecks at the servers when popular files (and more recently videos and movies in streaming services) are downloaded and shared in social

networks, producing high delays and degrading the performance of the networks.

In the same context, networks, such as Peer-to-Peer (P2P), emerge in a disruptive way to eliminate the need for central nodes and instead act together to complete tasks in common.

In a decentralized P2P network, a file is shared among the nodes that belong to the network, and all these nodes behave as servers and clients, sharing resources (memory, bandwidth, processing, and more).

In addition, in the case of the P2P-BitTorrent network studied in this article, the transmitted files are divided into small segments called chunks so that the nodes that contain the complete file are named seeds ( $x$ ), while the Nodes that do not contain the full chunks are known as leechers ( $y$ ).

Furthermore, while centralized networks aim to have distribution centers that can create conflicts of security, cost, stability, and other problems, P2P networks aim to expand to create reliable operations and make the nodes active in the network [7], however, in decentralized architectures apart from well-known security issues, such as worm distributions and viruses, the main problem is that the system operation relies heavily on nodes' collaboration to share their files and resources for the benefit of all users.

Although the phenomenon of collaboration has been largely overlooked through rules of connectivity, priorities and incentive mechanisms,

**Table 1.** P2P system states

State	Condition	$\tau$
<b>Penury</b>	$c x_\sigma > \mu(\eta x_\sigma + y_\sigma)$	$\mu(\eta x_\sigma + y_\sigma)$
<b>Abundance</b>	$c x_\sigma < \mu(\eta x_\sigma + y_\sigma)$	$c x_\sigma$

this paper studies the fact that in real P2P networks, there are certain nodes that since the first moment connect anonymously or are not disposed to share information with the other peers but they also download the available chunks using the network's resources, for these reasons, it is important to highlight the fact that for our paper we consider an initial state of collaboration ( $\sigma$ ) for every node in the network ( $v_0, v_1, v_2, \dots, v_{n-1}, v_n$ ), where this  $\sigma$  state is a binary variable that says if the node is sharing information ( $\sigma = \sigma_1 = 1$ ) or not ( $\sigma = \sigma_0 = 0$ ). In the same line, we consider that a non-collaborative peer ( $v_{\sigma_0}$ ) is not willing to share information, even if it has all the information ready to share, on the contrary, a collaborative peer ( $v_{\sigma_1}$ ) will try to share data through all the network during its lifespan.

It is important to notice that every leecher and every seed contain a  $\sigma$  that defines their state of collaboration and for the case of conversions from leechers to seeds, the new seeds will preserve the antique  $\sigma$  state as when they were leechers. For the previous reasons, the total number of seeds can be expressed as  $x_\sigma = x_{\sigma_1} + x_{\sigma_0}$ , which includes the collaborative and non-collaborative seeds due to the initial  $\sigma$  state, as well, the total number of leechers is expressed as  $y_\sigma = y_{\sigma_1} + y_{\sigma_0}$ .

In this sense, a collaborative initial state ( $\sigma$ ) allows greater fluidity in the processing of nodes for the transfer of information since it avoids the periods in which nodes have to determine priorities or apply incentive mechanisms in cooperation, allowing greater efficiency in nodes that require greater confidentiality without relegating them to the end of the process. Finally, a collaborative threshold ( $\zeta$ ) in the systems defines the maximum percentage of  $v_{\sigma_0}$  nodes in the environment, and once this percentage was reached, this type of nodes could not connect or would have to change their policies to be added as a  $v_{\sigma_1}$  or until some  $v_{\sigma_0}$

nodes leave the system. The main contributions of this paper are listed below:

- A Markovian model is proposed and solved through a Continuous Time Markov Chain (CTMC) that describes the fluid system of a P2P system.
- The model is validated through chain solutions (fluid model and simulation), describing the steps to follow and comparing each solution with the fluid model.
- A variation in the simulation of the system is proposed in order to study the impact of collaboration for leechers and seeds evolution in time.
- The behavior of a network that has variation in its collaboration is described.

The rest of the article is composed as follows: Section 2, presents the technologies in which P2P networks have been applied, as well as studies on the collaboration of nodes in this type of networks.

Section 3 details the dynamics of the P2P networks studied in this article through their fluid model, while Section 4 models the system through a CTMC and presents the algorithms that solve this chain through its fluid model and its simulation.

The final part of the article shows the relevant results from the CTMC solution and the impact of collaboration through the variation of  $v_{\sigma_0}$  nodes in the network, as well as conclusions and future work.

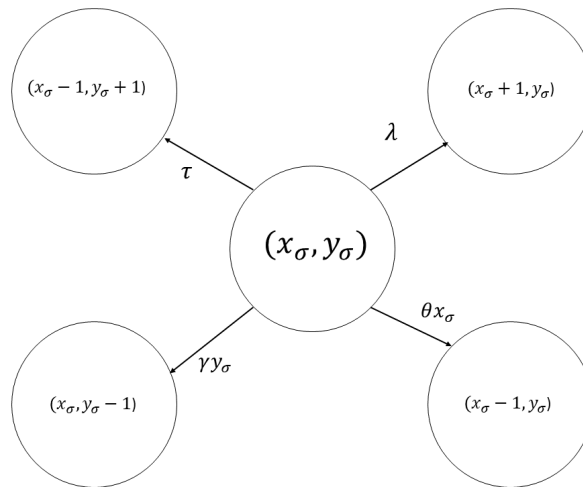
## 2 Related Work

The excessive growth in computing capabilities, the Internet, and new technologies such as IoT, AI, and VR are determining factors to search for other technologies and network architectures [8].

In the same line, P2P systems allow to implement decentralized file sharing by allowing all nodes to simultaneously act as servers and clients, increasing the resources in the network and, although P2P are still in constant development of cost reduction [14] in their infrastructure and detection of malicious nodes that may affect network processes [2], they are

**Table 2.** Solutions of the fluid model for stable conditions

State	Leechers	Seeds	$\gamma$
<b>Abundance</b>	$x_\sigma = \frac{\lambda}{\theta + c}$	$y_\sigma = \left(\frac{c}{\gamma}\right) \left(\frac{\lambda}{\theta + c}\right)$	$\gamma < \frac{\mu c}{c - \mu\eta}$
<b>Penury</b>	$x_\sigma = \frac{\lambda}{\theta + \mu\eta \left(1 - \frac{\mu}{\gamma - \mu}\right)}$	$y_\sigma = \frac{\mu\eta x_\sigma}{\gamma - \mu}$	$\gamma > \frac{\mu c}{c - \mu\eta}$



**Fig. 1.** CTMC of the P2P (BitTorrent) network

emerging as an alternative that could complement current environments having a significant peak in streaming services and with the future promise of a decentralized Internet as described by Philip Agre [1]. On the other hand, another important feature of P2P networks is the relationship that the peers maintain between them and the rules that they follow through the network.

In the same vein, few studies investigate the disposition of nodes to share information on the network and, although most refer to this process as collaboration, this collaboration can be approached from different points of view as shown by Chow [5, 4], who details how mobile nodes transmit information through caching techniques to reduce download times and to efficiently share information in the system, in the same way, Kumar [9] uses this procedure to vehicular networks

by modeling a discrete-time Markov chain that describes the initial, waiting, and updating states of these scenarios and the structures necessary for their operation.

The previous works manage the collaboration through the structure in which the nodes share due to their mobility and their physical capabilities in the transmission of data, however, there are other works that consider collaboration in P2P networks in terms of exchange not only of information but of energy as proposed by Amin [3] in its P2P electricity network in which there are collaborative and non-collaborative houses depending if they have solar panels and if they share them with the network.

It is important to note that there are interesting works whose objective is to study the collaboration in order to obtain certain parameters of the

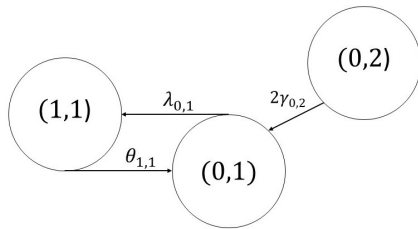


Fig. 2. Input and output transition rates for the state (0,1)

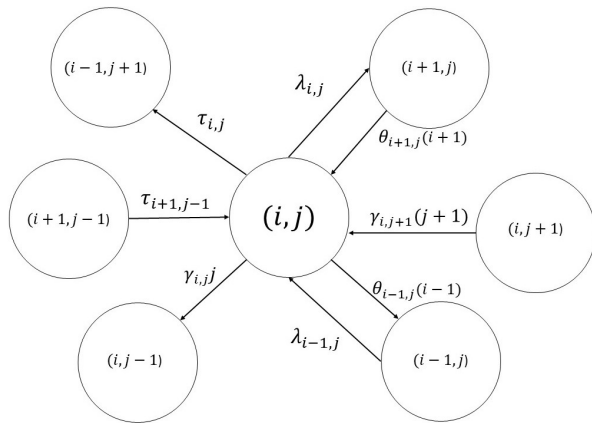


Fig. 3. Input and output transition rates for the state (i, j)

system, such as the one accomplished by Edgar Báez [6] for the calculation of collaborative and non-collaborative peers with different death rates, i.e., in this case the collaboration level is related to the average dwelling times inside the system, according to their priority scheme calculating latency, delay, successful downloads and percentage of blocked peers, or the study presented by Petrovic [11] in which a P2P system is analyzed in which there are different files of interest, then he details the download times and memory requirements for this type of scenario.

Distinctively to the articles described, this article focuses on the process of a Markovian analysis of the P2P system to study the collaboration among nodes considering that the peers enter anonymously (it is not possible to apply priority schemes) and determining if it will be collaborative ( $v_{\sigma_1}$ ) or non-collaborative ( $v_{\sigma_0}$ ) (through a proposed collaborative threshold ( $\zeta$ )) from the first moment that the node enters in the network and keeping it in that state until

its departure, finally, we analyze the impact of these initial states ( $\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_{n-1}, \sigma_n$ ) of collaboration through the evolution of the network on the average number of leechers and seeds.

Those adjustments allow us to improve the speed of sharing information and safeguard nodes' anonymity, which means a better performance of P2P networks applied to technologies such as IoT that connect a lot of devices through the same network (many times in real-time) avoiding larger control information frames both uploading and downloading data.

### 3 P2P Networks

As we mentioned before, P2P-BitTorrent networks are very useful for sharing trending information through a large number of nodes with better performance than centralized networks, moreover, the P2P behavior is caused by some parameters that will be detailed in this section, however, it is just as important how nodes connect to each other according to the following intrinsic rules of P2P mechanisms:

- Seeds do not connect to each other, because they already contain all the information available in the network, however, they can connect to any leecher to upload parts of the file.
- Leechers that connect to any seed can download all available chunks (if network rules allow it).
- Leechers will only connect to other leechers that have different chunks between them.

Likewise, the nodes of the system can download with a rate  $c$  and upload information with a rate  $\mu$ , having as a general (but not necessary) assumption:

$$c \geq \mu. \tag{1}$$

It is important to mention that bandwidths  $c$  and  $\mu$  will determine the rate in which a leecher becomes a seed ( $\tau$ ), in such a way that, if the system has enough resources, it means that the leechers are interacting with a high upload rate of chunks and this generates a greater available bandwidth, which means that

**Algorithm 1:** Markov chain algorithm for a P2P network

---

**Data:**  $\eta$ , iterations,  $\mu$ ,  $c$ , nodes,  $\lambda$ ,  $\theta$ ,  $\gamma$   
**Result:**  $x_\sigma$ ,  $y_\sigma$

```

1 tsim  $\leftarrow$  0;
2 list.add( $x_\sigma$ );
3  $x_\sigma \leftarrow$  1;
4  $y_\sigma \leftarrow$  0;
5 while  $i \leq$  iterations do
6   tmin, position  $\leftarrow$  min( $\eta$ ,  $\mu$ ,  $\lambda$ ,  $\theta$ ,  $\gamma$ );
7   tsim  $\leftarrow$  tsim + tmin;
8   if node  $\exists$  in list then
9     node.time  $\leftarrow$  node.time + tmin;
10  else
11    list.add( $x_\sigma$ , node.time);
12  switch position do
13    case 0 do
14       $x_\sigma = x_\sigma + 1$ ;
15    case 1 do
16       $x_\sigma = x_\sigma - 1$ ;
17    case 2 do
18       $y_\sigma = y_\sigma - 1$ ;
19    otherwise do
20       $x_\sigma = x_\sigma - 1$ ;
21       $y_\sigma = y_\sigma + 1$ ;
22 list_pr  $\leftarrow$  probabilities(list);
23 for node in list do
24    $x_\sigma = x_\sigma +$  node.probability  $\times$  node.leechers;
25    $y_\sigma = y_\sigma +$  node.probability  $\times$  node.seeds;

```

---

the maximum download rate will be given by the product between the system leechers and the download rate ( $c x_\sigma$ ).

Conversely, if the interaction of leechers is performed through a large number of repeated chunks (modeled by  $\eta$ ), bottlenecks could appear indicating that all available bandwidth is being consumed and causing a lower download rate in the network, therefore, the conversion of leechers to seeds will be modeled by the upload bandwidth, the number of seeds and the interaction between leechers ( $\mu(\eta x_\sigma + y_\sigma)$ ). Finally, the rate at which leechers are converted to seeds is defined as  $\tau$  and

is given by the following relationship:

$$\tau = \min(c x_\sigma, \mu(\eta x_\sigma + y_\sigma)), \quad (2)$$

where

$\tau$  = conversion rate ( $\forall \tau \in \mathbb{R}, \tau > 0$ ).  
 $c$  = downloading bandwidth ( $\forall c \in \mathbb{R}, c > 0$ ).  
 $\mu$  = uploading bandwidth ( $\forall \mu \in \mathbb{R}, \mu > 0$ ).  
 $\eta$  = sharing effectiveness ( $\forall \eta \in \mathbb{R}, 0 < \eta \leq 1$ ).  
 $x_\sigma$  = number of leechers ( $\forall x_\sigma \in \mathbb{N}, x_\sigma \geq 0$ ).  
 $y_\sigma$  = number of seeds ( $\forall y_\sigma \in \mathbb{N}, y_\sigma \geq 0$ ).

This system can be represented as a fluid model analyzed by D. Qiu [13, 12, 10] and developed for P2P-BitTorrent networks, and it can be expressed in a system of differential equations that are shown below in Equations 3 and 4:

$$\frac{d(x_{\sigma_1} + x_{\sigma_0})}{dt} = \lambda - \theta(x_{\sigma_1} + x_{\sigma_0}) - \tau, \quad (3)$$

$$\frac{d(y_{\sigma_1} + y_{\sigma_0})}{dt} = \tau - \gamma(y_{\sigma_1} + y_{\sigma_0}). \quad (4)$$

For stable conditions, in which a slight disturbance does not produce too disruptive effects on the system, we have:

$$\lambda - \theta x_\sigma - \tau = 0, \quad (5)$$

$$\tau - \gamma y_\sigma = 0. \quad (6)$$

Depending on the values of their bandwidths and output/input rates to the system, this type of network is classified into two states: penury and abundance. This relation is described in Table 1. For abundance state we substitute  $\tau = c x_\sigma$  in equations 5 and 6:

$$\lambda - \theta x_\sigma - c x_\sigma = 0, \quad (7)$$

$$c x_\sigma - \gamma y_\sigma = 0. \quad (8)$$

Clearing  $x_\sigma$  variable:

$$x_\sigma = \frac{\lambda}{\theta + c}, \quad (9)$$

$$x_\sigma = \frac{\gamma y_\sigma}{c}. \quad (10)$$

Then:

$$\frac{\gamma y_{\sigma}}{c} = \frac{\lambda}{\theta + c}.$$

Solving  $y_{\sigma}$ :

$$y_{\sigma} = \left(\frac{c}{\gamma}\right) \left(\frac{\lambda}{\theta + c}\right).$$

The same way is applied to find  $x_{\sigma}$  and  $y_{\sigma}$  values for  $\tau = \mu(\eta x_{\sigma} + y_{\sigma})$  in penury state, the range of values that are valid for the departure rate of the seeds ( $\gamma$ ) is obtained depending on the state in which the study is to be carried out, as shown in Table 2.

## 4 Mathematical Analysis

This paper studies the operation and collaboration of the P2P networks through a Markovian analysis in which the system can be represented as a homogeneous, irreducible, aperiodic CTMC with states  $(x_{\sigma}, y_{\sigma})$  positive recurring (ergodic chain) like the one shown in Fig. 1, where:

- $\lambda$  = leechers input rate ( $\forall \lambda \in \mathbb{R}, \lambda > 0$ ).
- $\theta$  = leechers output rate ( $\forall \theta \in \mathbb{R}, \theta > 0$ ).
- $\gamma$  = seeds output rate ( $\forall \gamma \in \mathbb{R}, \gamma > 0$ ).
- $\tau$  = conversion rate ( $\forall \tau \in \mathbb{R}, \tau > 0$ ).
- $x_{\sigma}$  = number of leechers ( $\forall x_{\sigma} \in \mathbb{N}, x_{\sigma} \geq 0$ ).
- $y_{\sigma}$  = number of seeds ( $\forall y_{\sigma} \in \mathbb{N}, y_{\sigma} \geq 1$ ).

CTMC has a space of valid states  $\Omega : \{(x_{\sigma}, y_{\sigma}) \mid x_{\sigma}, y_{\sigma} \in \mathbb{N}\}$ , such for the states  $(x_{\sigma}, y_{\sigma})$  it has the following transitions:

- There is a transition to state  $(x_{\sigma} + 1, y_{\sigma})$  with rate  $\lambda$  and denotes the arrival of a new leecher to the system.
- There is a transition to the  $(x_{\sigma} - 1, y_{\sigma})$  state with rate  $(\theta)(x_{\sigma})$  and denotes the output of a leecher from the system.
- There is a transition to the  $(x_{\sigma}, y_{\sigma} - 1)$  state with rate  $(\gamma)(y_{\sigma})$  and denotes the output of a seed from the system.
- There is a transition to the  $(x_{\sigma} - 1, y_{\sigma} + 1)$  state with rate  $(\tau)$  and denotes the conversion from a leecher to a seed.

---

### Algorithm 2: Simulation algorithm for a P2P network with collaboration variation

---

**Data:** chunks, threshold,  $\eta$ , iterations,  $\mu$ ,  $c$ , nodes,  $\lambda, \theta, \gamma$

**Result:**  $x_{\sigma}, y_{\sigma}$

```

1 tsim  $\leftarrow$  0;
2 list.add( $y_{\sigma}$ (ID, status, connections, chunks,  $\sigma_1$ ));
3 list.add( $x_{\sigma}$ (ID, status, connections, chunks,  $\sigma_1$ ));
4  $y_{\sigma} \leftarrow 1$ ;
5  $y_{\sigma_1} = 1$ ;
6  $y_{\sigma_0} = 0$ ;
7  $x_{\sigma} \leftarrow 0$ ;
8  $x_{\sigma_1} = 1$ ;
9  $x_{\sigma_0} = 0$ ;
10 while  $i \leq$  iterations do
11   event  $\leftarrow$  list.get_first();
12   tsim  $\leftarrow$  event.time();
13    $c_{\sigma_1}, c_{\sigma_0} \leftarrow$  conversions(tsim);
14    $y_{\sigma_1} \leftarrow y_{\sigma_1} + c_{\sigma_1}$ ;
15    $y_{\sigma_0} \leftarrow y_{\sigma_0} + c_{\sigma_0}$ ;
16    $x_{\sigma_1} \leftarrow y_{\sigma_1} - c_{\sigma_1}$ ;
17    $x_{\sigma_0} \leftarrow x_{\sigma_0} - c_{\sigma_0}$ ;
18   if event.type ==  $x_{\sigma}$  then
19     if event.action == birth then
20        $x_{\sigma} = x_{\sigma} + 1$ ;
21       event.action = death;
22       event.time = exponential(tsim,  $\theta$ );
23       search_tx(event, tsim);
24       event.time = exponential(tsim,  $\lambda$ );
25       if random(0, 1)  $\leq$   $\zeta$  then
26         list.add( $x_{\sigma}$ (ID, ...,  $\sigma_1$ ));
27          $x_{\sigma_1} \leftarrow x_{\sigma_1} + 1$ ;
28       else
29         list.add( $x_{\sigma}$ (ID, ...,  $\sigma_0$ ));
30          $x_{\sigma_0} \leftarrow x_{\sigma_0} + 1$ ;
31     else
32        $x_{\sigma} = x_{\sigma} - 1$ ;
33       event.clearconnections();
34       if event.collaborative == True then
35          $x_{\sigma_1} \leftarrow x_{\sigma_1} - 1$ ;
36       else
37          $x_{\sigma_0} \leftarrow x_{\sigma_0} - 1$ ;
38   else
39     if  $y_{\sigma} > 1$  then
40        $y_{\sigma} \leftarrow y_{\sigma} - 1$ ;
41       event.clearconnections();
42       if event.collaborative == True then
43          $y_{\sigma_1} = y_{\sigma_1} - 1$ ;
44       else
45          $y_{\sigma_0} = y_{\sigma_0} - 1$ ;
46     else
47       event.time = exponential(tsim,  $\gamma$ );

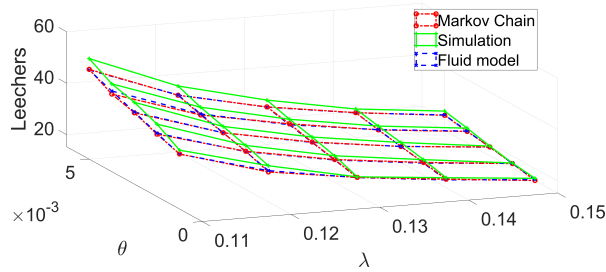
```

---

The initial state is  $(0, 1)$ , where there is no leechers asking for information and there is just one seed, for the above we cannot hope for a

**Table 3.** Table of parameters and values utilized

Parameter	Value
$\lambda$	[0.11, 0.12, 0.13, 0.14, 0.15]
$\theta$	[0.001, 0.002, 0.003, 0.004, 0.005]
$\gamma$	[0.0009, 0.0012, 0.0015, 0.0018, 0.0021]
$\zeta$	[0.2, 0.4, 0.6, 0.8, 1.0]
<b>iterations</b>	50,000
$c$	0.002
$\mu$	0.00125

**Fig. 4.** Number of leechers in the system for the fluid model and the Markovian model in  $\zeta = 1$ 

seed departure (because there must be at least one seed in the system) or a leecher departure but we can see that if state  $(0, 2)$  presents a seed departure, we can return to  $(0, 1)$  as is shown in Fig. 2. In the same way, we can generalized arrives and departures from state  $(i, j)$  as in the Fig. 3 and for steady state this chain can be solved by equalizing their output and input transition rates, that is, by a flow balance Eqn. 11:

$$p_{0,1}(\lambda_{0,1}) = p_{1,1}\theta_{1,1} + 2p_{0,2}\gamma_{0,2}, \quad (11)$$

$$p_{1,1}(\lambda_{1,1} + \tau_{1,1} + \theta_{1,1}) = p_{0,1}\lambda_{0,1} + 2p_{2,1}\theta_{2,1}. \quad (12)$$

$$\vdots$$

$$p_{15,16}(\lambda_{15,16} + \dots + \tau_{15,16}) = p_{14,16}\lambda_{14,16} + \dots + \tau_{16,15}. \quad (13)$$

$$\vdots$$

$$p_{i,j}(\lambda_{i,j} + \dots + \tau_{i,j}) = p_{i-1,j}\lambda_{i-1,j} + \dots + \tau_{i+1,j-1}. \quad (14)$$

Once the Markovian model was described, the CTMC was solved numerically by modeling the jumps in the chain, creating exponential times corresponding to each transition rate as it is shown in Algorithm 1. After the numerical solution for the fluid model, we solve the chain by means

of the Algorithm 2 simulating the events and characteristics of the system and applying the collaborative threshold  $\zeta$  in the system.

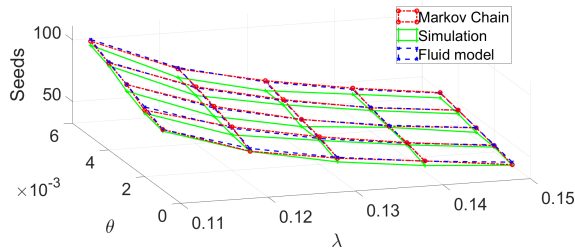
Before any variation, our simulation was compared with the numerical solution to validate the data obtained in the simulation. A more detailed description of algorithms 1 and 2 is found below, in which, we take into consideration the collaboration between the nodes, remembering that for our paper we consider an initial state  $\sigma$  for every node in the network.

Furthermore, we remark and show the behavior of the state  $\sigma_i$  for a node  $i$  from the first moment that it gets into the network to the last moment it leaves the system. For the above, we remark that a  $v_{\sigma_0}$  peer has an initial collaboration state  $\sigma = 0$ , which means that, this node will not share information, contrary, we consider a  $v_{\sigma_1}$  peer when it has  $\sigma = 1$ .

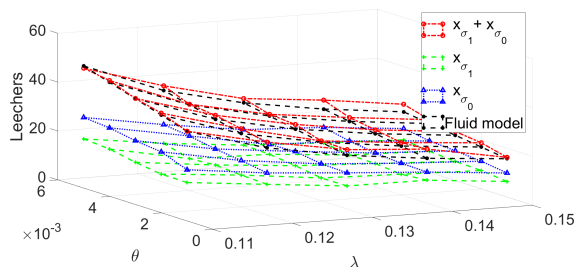
#### 4.1 CTMC Numerical Solution

The process of Algorithm 1 is detailed below:

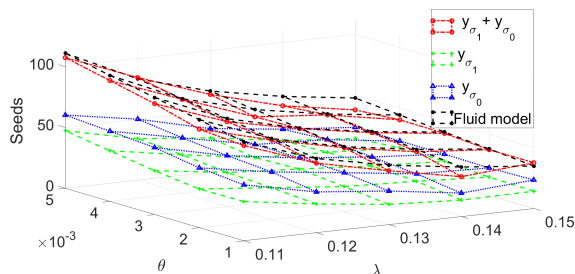
- As a first step, the rates, bandwidths, and the number of iterations detailed in the Algorithm 1 are entered. However, for the simulation method, it will also be required to detail the number of chunks and the parameter  $\zeta$  that will determine the collaboration between the present nodes in the network.
- In the next stage, the simulation time (tsim) and the number of leechers ( $x_\sigma$ ) are initialized to zeros, and a seed is added to the list.
- From this moment on, the subsequent steps will be repeated during the indicated iterations. Next, the function  $\min()$  is executed, which will create the exponential times corresponding to each possible state that can be accessed from the state  $(x_\sigma, y_\sigma)$  and which will later determine the smallest time (tmin) and its position (position), taking into account that this shortest time corresponds to the time to the nearest event.



**Fig. 5.** Number of seeds in the system for the fluid model and the Markovian model with  $\zeta = 1$



**Fig. 6.** Collaborative ( $x_{\sigma_1}$ ) and non-collaborative ( $x_{\sigma_0}$ ) leechers calculated by the simulations ( $\zeta = 0.4$ ,  $\gamma = 0.0009$ ) compared to the ideal collaborative scenario ( $\zeta = 1$ ,  $\gamma = 0.0009$ )



**Fig. 7.** Collaborative ( $y_{\sigma_1}$ ) and non-collaborative ( $y_{\sigma_0}$ ) seeds calculated by the simulations ( $\zeta = 0.4$ ,  $\gamma = 0.0009$ ) compared to the ideal collaborative scenario ( $\zeta = 1$ ,  $\gamma = 0.0009$ )

- Once the minor time has been identified, it proceeds to add it to the simulation time, and this time would correspond to the time that the state  $(x_{\sigma}, y_{\sigma})$  remained before jumping to the new state.
- Subsequently, the transitions described above are considered cases in which the cases of conversion, departure of leechers/seeds from the system, and the arrival of leechers are considered.

- As a final step, the function probabilities() is executed, which will calculate the steady probabilities (list\_pr) corresponding to each state present in the solution equations ( 15 and 16), to finally obtain the average number of leechers and seeds equations(17 and 18):

$$P(X_i) = \frac{\sum_{i=0}^{\infty} t(x_i)}{t_{\text{simulation}}}, \quad (15)$$

$$P(Y_i) = \frac{\sum_{i=0}^{\infty} t(y_i)}{t_{\text{simulation}}}, \quad (16)$$

$$P(\bar{X}) = \sum_{i=0}^{\infty} P(X_i)X_i, \quad (17)$$

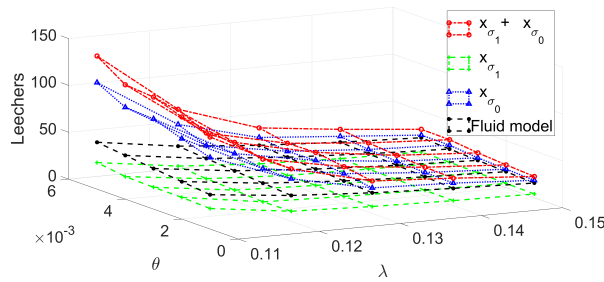
$$P(\bar{Y}) = \sum_{i=0}^{\infty} P(Y_i)Y_i. \quad (18)$$

#### 4.2 Discrete Event Simulator

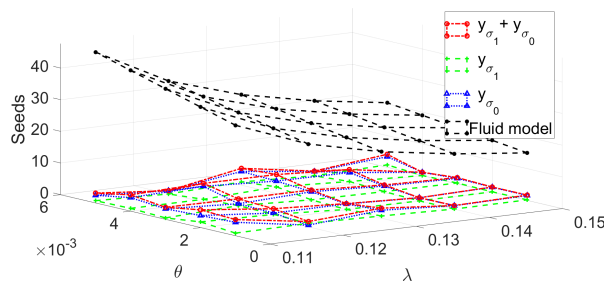
As with the numerical solution algorithm, the corresponding steps of the algorithm are detailed by Algorithm 2:

- At first instance, the input rate of leechers ( $\lambda$ ), the output rates of leechers and seeds ( $\theta$ ,  $\gamma$ ), as well as the upload and download bandwidths ( $c$ ,  $\mu$ ), the number of iterations (iterations), and the variable  $\eta$  are acquired. Finally, the average number of leechers and seeds in the system will be obtained.
- In the next stage, the simulation time (tsim) is initialized, then a seed and a leecher are added to the collaborative mode list. In other words, these nodes, in addition to downloading information, will collaborate by actively sharing them throughout the network. In order to have a count of the collaboration of the nodes, the counters of collaborative nodes ( $x_{\sigma_1}$ ,  $y_{\sigma_1}$ ) and non-collaborative nodes ( $x_{\sigma_0}$ ,  $y_{\sigma_0}$ ) are also initialized.

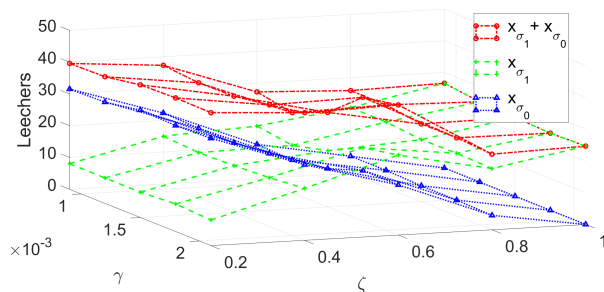




**Fig. 8.** Collaborative ( $x_{\sigma_1}$ ) and non-collaborative ( $x_{\sigma_0}$ ) leechers calculated by the simulation compared to the fluid model ( $\zeta = 0.2, \gamma = 0.0021$ )



**Fig. 9.** Collaborative ( $y_{\sigma_1}$ ) and non-collaborative ( $y_{\sigma_0}$ ) seeds calculated by the simulation compared to the fluid model ( $\zeta = 0.2, \gamma = 0.0021$ )



**Fig. 10.** Impact of the variation of  $\zeta$  and the rate  $\gamma$  on the number of leechers of a P2P system ( $\lambda = 0.12, \theta = 0.003$ )

- The following steps will be repeated for the number of iterations entered. Next, the first node in the list (`list.get_first()`) will be taken, and its time (`event.time()`) will add that time to the simulation time.
- Subsequently, the number of conversions() from leechers to seeds in that time (if any) will be obtained and divided between collaborative ( $c_{\sigma_1}$ ) and non-collaborative ( $c_{\sigma_0}$ ) conversions.

- As the next step, the current event will be evaluated, and depending on its type and action to be carried out, following possibilities will be obtained:

- If the type of the event (`event.type`) is a leecher and its action is a birth, then a unit is added to the leechers counter (collaborative or non-collaborative as the case may be) and a new time (`event.time`) is created for it, changing its action to death (`event.action`).

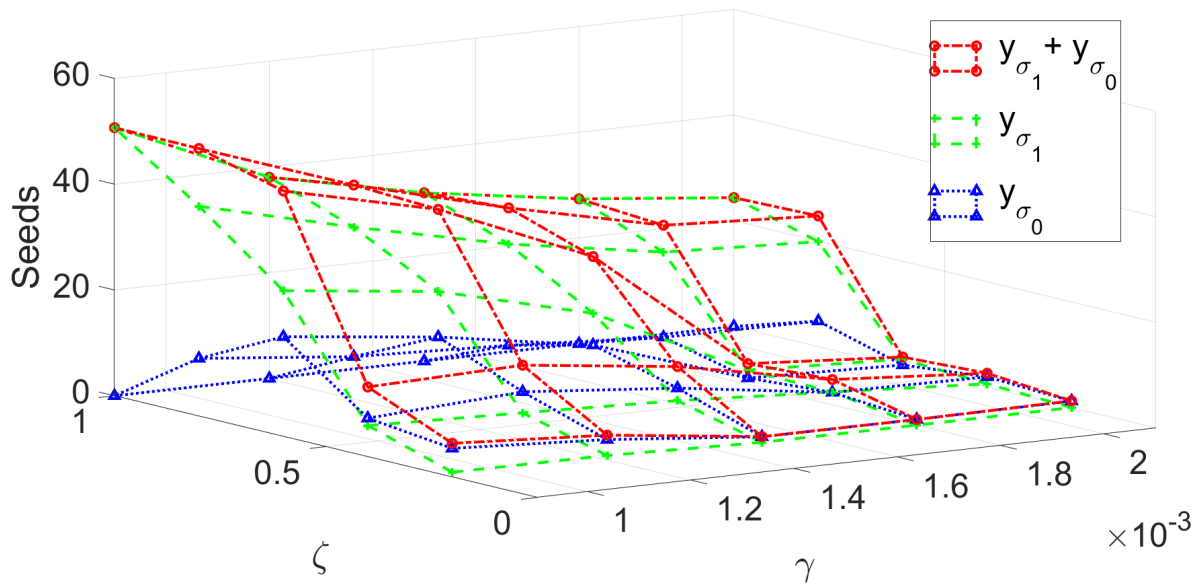
Then, the possible providers of information in the network are searched for (`search_tx()`), and a new birth is created, which, depending on the random value (`random(0, 1)`) generated in comparison with the collaboration, this comparison determine if rebirth will be collaborative or non-collaborative.

- If the type of the event is a leecher but has a death action, then the leecher's counter is decreased by one unit ( $x_{\sigma_1}$  or  $x_{\sigma_0}$ , as the case may be), and the upload and download connections of the node are eliminated (`event.clearconnections()`).

- Finally, if the event type is a seed with a death action, then the seed counter is decreased by one unit ( $y_{\sigma_1}$  or  $y_{\sigma_0}$ , as the case may be), and the upload and download connections of the node are eliminated `event.clearconnections()`. It should be noted that there must always be at least one seed in the system, so if there is a death and only one seed in the system, a new death time for this node is generated.

- At the end of the iterations, the average number of leechers and seeds is obtained through the simulation, as well as the collaborative and non-collaborative nodes.

Under the collaborative circumstances described, we assume the same input ( $\lambda$ ) and output ( $\theta, \gamma$ ) rates for collaborative peers and non-collaborative peers in order to observe the queuing of leechers and seeds when they are not willing to share information, this behavior is considered with the collaborative threshold  $\zeta$  applied to the simulation when a peer is created, then, for every new node considered in the



**Fig. 11.** Impact of the variation of  $\zeta$  and the  $\gamma$  rate on the number of *seeds* of a P2P system ( $\lambda = 0.12$ ,  $\theta = 0.003$ )

simulation we decided the state  $\sigma$  by means of the comparison between a determined value of  $\zeta$  ( $0 < \zeta < 1$ ); if  $\xi \leq \zeta$  then the new peer will have  $\sigma = 1$  and it will be collaborative with the other nodes otherwise it will be non-collaborative with  $\sigma = 0$ . It is important to mention that when  $\zeta = 1$  we have an ideal P2P environment in which all peers distribute information in the network, on the other hand, if  $\zeta = 0$  we will have a centralized network in which only the initial node (that owns the file) could be capable of transmitting information to the other peers (Client-Serve architecture).

## 5 Results

Once the chain was solved, the fluid model was compared with the Markovian model through the DES of the system and the numerical solution of the Markov chain with  $\zeta = 1$ . The results of these comparisons are shown in the figures 4 and 5. At the same time, the parameters used for these solutions are indicated in Table 3. Figures 4 5 provide a validation of the proposed CTMC and the methods used for its solution. For this reason, the following subsection will show the results obtained from simulating the system with a collaborative threshold different from unity ( $0 < \zeta < 1$ ).

### 5.1 Collaboration in P2P Environments

Ideally, all peers are willing to cooperate through the P2P network, however, in real environments, this could be different. The modeling referring to the P2P network studied in this work suggests another that allows glimpsing system's operation varying the  $\zeta$  parameter in the network (Algorithm 1).

Figures 6 and 7 show that although the collaboration in this experiment was lowered, the values of leechers and seeds remained very close to the values calculated by the fluid model through the variation of the  $\lambda$  and  $\theta$  rates with a  $\gamma$  value of 0.0009. The preceding shows that although many non-collaborative nodes exist in the system, an equilibrium is maintained that allows operation very close to the ideal operation.

However, figures 8 and 9 show a scenario in which collaboration between nodes is further reduced and output rate of the seeds ( $\gamma$ ) increases, resulting in a network unable to provide the information in an effective way, producing a higher number of leechers in the queue and a very low number of seeds. Consequently, it can be affirmed that the level of collaboration directly impacts the conversion of leechers to seeds.

Hence, figures 8 and 9 result from variations in the arrival and departure rates of the nodes ( $\lambda$  and  $\theta$  respectively) while the departure rate of the seeds ( $\gamma$ ) was constant with the aim of observe the bearing of the network from the behavior of the leechers, then, it is important to study the system through the collaboration sweep varying  $\gamma$  and holding  $\lambda$  and  $\theta$  constant.

For the above, Figures 10 and 11 show that the lower the collaboration, the greater the number of leechers in the queue (since leechers only depend on their arrival rate), and they will mostly be non-collaborative, which would cause a low number of seeds in the system (the number of conversions decreases). On the contrary, the greater the collaboration, the number of leechers and seeds will remain stable.

## 6 Conclusions and Future Work

This work developed a Markovian model that describes a P2P network, besides, we solve the model numerically and the simulation of the system, validating the solutions' results with pre-existing fluid models.

On the other hand, necessary adjustments were made to achieve variation in collaboration between the nodes of the network through the parameter  $\zeta$ , taking into account real environments in which it is not possible to control all the peers and their collaboration. Furthermore, we propose to include the initial state of collaboration  $\sigma$  in the simulation to provide a better approach to real environments and then estimate the performance in the network according to this control variable.

On the other hand, despite the adjustments applied allowing nodes to hold their anonymity, the parameter  $\zeta$  defines the total percentage of  $\sigma_0$  nodes authorized in the network, corroborating that the smaller number of  $x_{\sigma_0}$  nodes in the environment the greater number of  $y_{\sigma_1}$  are created and then, the complete file can be shared through all the nodes more efficiently.

For the above, it can be affirmed that a P2P network can work efficiently, finding the balance of its rates even if its collaboration is not maximum ( $\zeta = 1$ ) and thus maintaining a performance similar to that presented by the fluid model.

However, it is clear that if the balance as mentioned above is not achieved, a null creation of collaborative leechers ( $x_{\sigma_1}$ ) can be obtained, which in turn translates into a very low number of seeds and, therefore, an inefficient P2P network in which it is complicated to get the information within the network. Therefore, it would be convenient and interesting to calculate the collaboration balance for future work and determine the impact found through  $\zeta$  directly in the fluid model.

## Data Availability

All experiments and data used in this work are available if required.

## Acknowledgments

This research was funded by Instituto Politécnico Nacional grant number SIP 20231239 and CONACYT.

## References

1. **Agre, P. E. (2003)**. P2P and the promise of internet equality. *Communications of the ACM*, Vol. 46, No. 2, pp. 39–42. DOI: 10.1145/606272.606298.
2. **Al-Otaiby, N., Alhindi, A., Kurdi, H. (2022)**. Antitrust: An ant-inspired trust management system for peer-to-peer networks. *Sensors*, Vol. 22, No. 2, pp. 533. DOI: 10.3390/s22020533.
3. **Amin, W., Huang, Q., Afzal, M., Khan, A. A., Umer, K., Ahmed, S. A. (2020)**. A converging non-cooperative and cooperative game theory approach for stabilizing peer-to-peer electricity trading. *Electric Power Systems Research*, Vol. 183, pp. 106278. DOI: 10.1016/j.epsr.2020.106278.
4. **Chi-Yin, C., Hong-Va, L., Chan, A. T. S. (2007)**. GroCoca: Group-based peer-to-peer cooperative caching in mobile environment. *IEEE Journal on Selected Areas in Communications*, Vol. 25, No. 1, pp. 179–191. DOI: 10.1109/jsac.2007.070118.

5. **Chow, C. Y., Leong, H. V., Chan, A. (2004).** Peer-to-peer cooperative caching in mobile environments. Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, pp. 528–533. DOI: 10.1109/icdcs.2004.1284083.
6. **Esquivel, E. E. B., Rivero-Angeles, M. E., Orea-Flores, I. Y. (2014).** Performance analysis of bittorrent-like P2P networks for video streaming services at the chunk level. Proceedings of the IEEE 28th International Conference on Advanced Information Networking and Applications, pp. 806–812. DOI: 10.1109/AINA.2014.98.
7. **Karaata, M., Al-Mutairi, A., Alsubaihi, S. (2022).** Multipath routing over star overlays for quality of service enhancement in hybrid content distribution peer-to-peer networks. IEEE Access, Vol. 10, pp. 7042–7058. DOI: 10.1109/access.2021.3139936.
8. **Kumar, D., Pandey, M. (2022).** An optimal load balancing strategy for P2P network using chicken swarm optimization. Peer-to-Peer Networking and Applications, Vol. 15, No. 1, pp. 666–688. DOI: 10.1007/s12083-021-01259-3.
9. **Kumar, N., Lee, J. H. (2014).** Peer-to-peer cooperative caching for data dissemination in urban vehicular communications. IEEE Systems Journal, Vol. 8, No. 4, pp. 1136–1144. DOI: 10.1109/jsyst.2013.2285611.
10. **Ma, Z., Qiu, D. (2009).** A novel optimistic unchoking algorithm for bittorrent. Proceedings of the 6th IEEE Consumer Communications and Networking Conference, pp. 1–4. DOI: 10.1109/CCNC.2009.4784859.
11. **Petrovic, S., Brown, P. (2009).** Large scale analysis of the eDonkey P2P file sharing system. Proceedings of the IEEE International Conference on Computer Communications, pp. 2746–2750. DOI: 10.1109/infcom.2009.5062224.
12. **Qiu, D., Sang, W. (2008).** Global stability of peer-to-peer file sharing systems. Computer Communications, Vol. 31, No. 2, pp. 212–219. DOI: 10.1016/j.comcom.2007.08.012.
13. **Qiu, D., Srikant, R. (2004).** Modeling and performance analysis of bittorrent-like peer-to-peer networks. ACM SIGCOMM Computer Communication Review, Vol. 34, No. 4, pp. 367–378. DOI: 10.1145/1030194.1015508.
14. **Singh, S. K., Kumar, C., Nath, P. (2022).** Resource-cardinality based scheme to reduce resource lookup cost in structured P2P networks. Wireless Personal Communications, Vol. 125, No. 4, pp. 3351–3377. DOI: 10.1007/s11277-022-09714-x.

Article received on 16/10/2023; accepted on 21/11/2023.

\* Corresponding author is Mario E. Rivero-Angeles.