Submitted version

Peer-to-Peer Networking and Applications manuscript No.
(will be inserted by the editor)

Analysis of minimum distribution time of tit-for-tat-based P2P file distribution: Linear programming based approach

Masahiro Sasabe

Received: date / Accepted: date

Abstract In recent years, software update has become one of the essential functions in the current Internet to maintain devices safe from a various types of attacks. If the number of software users is enormous, distribution servers tend to be bottlenecks, due to the access concentration. To alleviate this problem, several systems (e.g., Windows update) have started applying peer-to-peer (P2P) file distribution where clients (peers) assist the distribution by uploading retrieved fragments of the file (i.e., pieces) to others. However, it has been pointed out that many peers tend to be free riders, which are not willing to upload pieces to others, so as to save their upload capacity. Tit-for-tat (TFT) strategy in game theory is one of the practical mechanisms to build reciprocity between each pair of peers. In this paper, we first develop a linear program (LP) as a tool to analyze the minimum distribution time of the TFT-based P2P file distribution under arbitrary upload capacity distribution. Focusing on the fact that access links can be categorized into multiple classes in the current Internet, we further extend the general formulation to multi-class formulation, which does not depend on the system scale. Through numerical results, we reveal the fundamental characteristics of the TFT-based P2P file distribution.

Keywords Peer-to-peer (P2P) file distribution · tit-for-tat (TFT) strategy · minimum distribution time · linear programming (LP)

This research was partly supported by JSPS KAKENHI (19H01103 and 19KT0045), Japan.

Masahiro Sasabe
Graduate School of Science and Technology,
Nara Institute of Science and Technology,
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
Tel.: +81-743-5361
Fax: +81-743-5369
E-mail: m-sasabe@ieee.org

1 Introduction

In recent years, software update has become one of the essential functions in the current Internet to maintain devices safe from a various types of attacks. If the software is proliferated as in operating systems (OS) (e.g., Windows, macOS, and Linux), distribution servers tend to be bottlenecks, due to the access concentration. In case of software update, periodical update (e.g., update per month) tends to be required to fix vulnerabilities found so far, which causes steep increase of access requests just after the new release (i.e., *flash crowd*) [1].

To alleviate the access concentration to the distribution servers, several systems (e.g., Windows Update) have started applying the peer-to-peer (P2P) assisted distribution [2]. In the P2P file distribution, the server divides the entire file into lots of small fragments (i.e., pieces or chunks) and sends them to user devices called *peers*. Each peer can retrieve pieces from the server as well as other peers, which contributes to reduction of load on the server. Note that the performance of such P2P file distribution systems will be highly affected by peers' behavior, i.e., cooperative/uncooperative to providing pieces to others.

Each peer is an individual user's device and consumes its own upload capacity to upload pieces to others. As a result, it has been pointed out that most peers tend to be free riders, which only download from others without uploading to others, so as to avoid degradation of quality of other network services running on its own device and/or additional monetary cost under the access fee plan depending on the usage [3]. BitTorrent [4], one of the most famous P2P file distribution systems, adopts a game theoretic approach called the tit-for-tat (TFT) strategy to achieve mutual reciprocity between any pair of peers [5].

Recently, several studies have analyzed the optimality of the TFT-based P2P content distribution from various aspects [6–9]. These studies formulated integer linear programs (ILPs) to determine the optimal piece flow for the TFT-based content distribution with different objectives. Note that the piece flow is the information that describes the transfer timing of pieces between nodes (i.e., server or peer). They obtained the optimal piece flow by solving the formulated ILPs using the existing solver (i.e., CPLEX [10]) and analyzed the obtained optimal flow to reveal the mechanism yielding the optimal piece flow (e.g., peer and piece selection at the server and peers). The piece-based precise analysis is the strength of these approaches while the complexity of ILPs hinders the direct analysis of large-scale systems.

As for the P2P file distribution without the TFT strategy, the minimum distribution time was explicitly derived using the fluid model under the assumption where all peers simultaneously finish file downloading [11]. In this paper, inspired by this approach, we formulate a linear program (LP) to minimize the file distribution time under the assumption where all peers finish file downloading at the same time. Although both LP and ILP are NP-hard, it is prone to be much easier to solve LP than ILP because of its continuous solution space [12]. In fact, we will show the proposed LP-based approach can directly solve the large scale systems with more than thousand nodes.

The main contributions of the manuscript are following.

1. An LP is developed as a tool to analyze the minimum distribution time of the TFT-based P2P file distribution under arbitrary upload capacity distribution.
2. By considering the fact that access links can be categorized into multiple classes in the current Internet, we further extend the general formulation to multi-class formulation whose problem size does not depend on the system scale, i.e., the number of peers.
3. Through numerical evaluations using CPLEX, we show the LP-based approach can analyze systems with more than thousand peers, which is much scalable than the existing ILP-based approach that can support less than tens of peers.
4. Through the analysis of the optimal solutions under two-class upload capacity scenarios (i.e., high-speed peers and low-speed ones), we reveal the fundamental characteristics of the optimality of the TFT-based P2P file distribution: (a) scalability to the number of peers, (b) impact of fraction of free riders, (c) impact of fraction of high-speed peers and capacity difference between high-speed and low-speed peers, (d) impact of upload capacity of server, and (e) impact of network partitioning.
5. The effectiveness of the TFT-based P2P file distribution is also confirmed through numerical results under practical scenario based on the number and distribution of fixed-line broadband service subscriptions in Japan.

The remaining part is organized as follows. Section 2 gives related work. After introducing the system model in Section 3, we formulate the LP to minimize the file distribution time of the TFT-based P2P file distribution in Section 4. Section 5 demonstrates the fundamental characteristics of the minimum file distribution time of the TFT-based P2P file distribution through numerical results. Finally, we give the conclusions and future work in Section 6.

2 Related work

Compared with the performance analysis of the conventional client–server file distribution, that of the P2P file distribution has been challenging, due to its complex features such as selfish/rational behavior of individual peers (e.g., free riding [3] and departure just after file downloading [13]) and its countermeasures (e.g., TFT strategy), distributed control, and peer churn [14].

There are extensive studies on the performance analysis of the P2P file distribution in different approaches: queuing models [15–17], fluid models [11, 18–23], discrete-time model [24–26], and ILP models [6–9].

From the viewpoint of the types of P2P file distribution, the initial studies mainly focused on the cooperative models where all peers are altruistically provide retrieved pieces with other peers [11, 15, 16, 19]. However, it has been pointed out that such cooperative models are not practical and many peers tend to be free riders in actual P2P file distribution systems [3]. BitTorrent is one the most successful P2P file distribution systems to alleviate the free-riding problem by introducing the TFT strategy into piece exchange between peers. Some of the existing studies analyzed the performance of BitTorrent(-like) systems [17, 20–22, 25–27]. BitTorrent does not include the TFT strategy but also other complicated mechanisms (e.g., optimistic unchoking). Since these mechanisms are intuitively designed and introduced, there are also several studies on analyzing the optimality of the TFT-based P2P file distribution to reveal the potential of the TFT strategy itself [6–9, 24].

In [24], the authors analyzed the minimum distribution time of the TFT model and compared with those of conventional client–server model and altruistic model. Their discrete-time and piece-level analysis relied on the assumptions where all nodes have the same upload capacity and all the pieces are of equal size. The optimality of the TFT-based P2P content distribution under heterogeneous upload capacity distribution among peers was also analyzed [6–9]. These studies developed ILP models to reveal the optimal piece-flow in terms of file distribution [6], video/audio streaming [7], logical P2P topology [8], and locality-awareness [9]. Although these ILP models enabled the precise piece-level analysis, they could not directly analyze the large-scale systems, due to high computation complexity.

Inspired by the fluid-model based analysis of the minimum distribution time of cooperative P2P file distribution [11], we newly develop LP model to minimize the distribution time of the TFT-based P2P file distribution under the assumption that all peers simultaneously finish file downloading. Although LP is also NP-hard as ILP, it is prone to be much easier to be solved because of its continuous solution space [12].

3 System model

We consider a TFT-based P2P distribution system that consists of one server, denoted by S, and N_{P} ($N_{\mathrm{P}} \geq 1$) peers, denoted by $\mathcal{N}_{\mathrm{P}}$. We also define the set of nodes (i.e., server or peers) as $\mathcal{N} = \{\mathrm{S}\} \cup \mathcal{N}_{\mathrm{P}}$. Note that the one server can also be regarded as a cluster of multiple servers. At the initial state ($t = 0$), only the server S has a file of size F ($F > 0$). On the other hand, all the peers initially do not have any portion (piece) of the file. They can obtain pieces from the server as well as the other peers. If a peer completes file downloading and has the whole file, it is called a *seed*. Otherwise, it is called a *leecher*.

Each node $i \in \mathcal{N}$ has an upload capacity U_i ($U_i \geq 0$), with which the node i can transmit bits at the maximum. Since the download capacity tends to be much larger than the upload capacity in the current Internet, we assume that the upload capacity of each node can be the bottleneck.

As in [11], we consider a fluid model where a leecher can replicate and forward a bit immediately after its reception. This assumption plays a key role of modeling the optimal rate allocation in the TFT-based P2P file distribution.

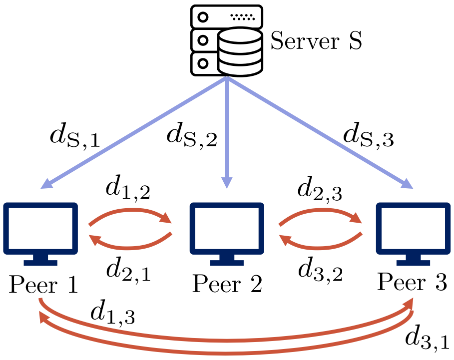


Fig. 1: Considered system configuration and download rate between nodes ($N_{\mathrm{P}} = 3$).

Fig. 1 illustrates the system configuration considered in this paper. There are N_{P} application-level multicast trees, each of which has the server as its root, one leecher $i \in \mathcal{N}_{\mathrm{P}}$ directly connected to the root, and the other $N_{\mathrm{P}}-1$ leechers connected to the leecher i. We focus on the direct transfer from the server to each leecher and the next-hop transfer from one leecher to others.

The main goal is to find the minimum file distribution time T_{min} and the corresponding download rate profile, $\boldsymbol{d} = (d_1, \ldots, d_{N_{\mathrm{P}}})$, under the assumption that all the peers simultaneously finish file downloading, as in [11]. Since all the peers finish file downloading at the same time, they can always be regarded as leechers in our analysis. Here, d_i ($d_i \geq 0$) is the download rate at which the leecher $i \in \mathcal{N}_{\mathrm{P}}$ downloads fresh content from the server and other leechers:

$$d_i = d_{\mathrm{S},i} + \sum_{j \in \mathcal{N}_{\mathrm{P}} \setminus \{i\}} d_{j,i}, \tag{1}$$

where $d_{i,j}$ ($i \in \mathcal{N}, j \in \mathcal{N}_{\mathrm{P}}, i \neq j, d_{i,j} \geq 0$) is the download rate from node i to node j. Since we assume that all the peers simultaneously finish file downloading, d_i will be identical, i.e., d ($d > 0$). Different from the system model in [11], our model requires for each pair of leechers i and j ($i, j \in \mathcal{N}_{\mathrm{P}}, i \neq j$) to exchange the same amount of data, i.e., $d_{i,j} = d_{j,i}$, under the TFT constraint, which makes the analysis more complicated.

So far we mainly focus on the rate allocation among nodes. In case of the file distribution, we also need to ensure that each leecher can retrieve fresh content from the server and other leechers. This will be satisfied in a similar way to that in [11]. Fig. 2 illustrates the initial bit-level flow of the leecher 1 when $N_{\mathrm{P}} = 3$. The serer S divides the file into N_{P} different pieces, each of which will have different size depending on $d_{\mathrm{S},i}$. The leecher 1 starts retrieving the beginning of the piece 1 from the server S at the rate of $d_{\mathrm{S},1}$. It immediately forwards the retrieved portion of the piece 1 to leecher 2 (resp. leecher 3) at the rate of $d_{1,2}$ (resp. $d_{1,3}$) while satisfying $d_{1,2} \leq d_{\mathrm{S},1}$ (resp. $d_{1,3} \leq d_{\mathrm{S},1}$). At this time, the leecher 1 also retrieves the piece 2 (resp. piece 3) from the

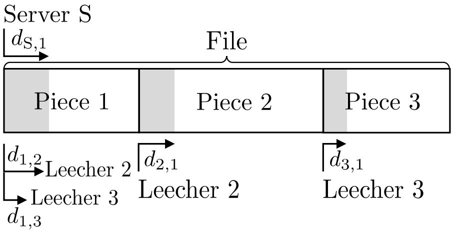


Fig. 2: Bit-level flow of leecher 1 at the initial state ($N_\mathrm{P} = 3$).

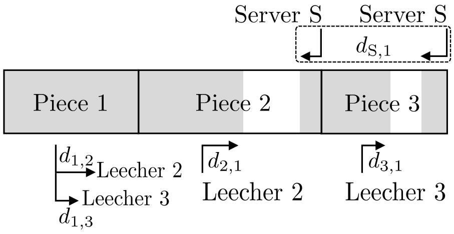


Fig. 3: Bit-level flow of leecher 1 after the complete download of allocated piece 1 in case of cooperative P2P distribution (When the server S's upload capacity is abundant compared with the upload capacities of peers, $N_\mathrm{P} = 3$).

leecher 2 (resp. leecher 3) at the rate of $d_{2,1}$ (resp. $d_{3,1}$) where $d_{2,1} = d_{1,2}$ (resp. $d_{3,1} = d_{1,3}$) should hold under the TFT constraint.

For simplicity, suppose all peers $i \in \mathcal{N}_\mathrm{P}$ have the same upload capacity $U_i = U$. If the server S's upload capacity becomes bottleneck, $d_{i,j} = d_{\mathrm{S},i}$, $d_{i,j} = d_{j,i}$, and $\sum_{j\in\mathcal{N}_\mathrm{P}\setminus\{i\}} d_{j,i} \leq U$ $(i,j \in \mathcal{N}_\mathrm{P}, i \neq j)$ are satisfied. Since $\sum_{i\in\mathcal{N}_\mathrm{P}} d_{\mathrm{S},i} = U_\mathrm{S}$, each peer $i \in \mathcal{N}_\mathrm{P}$ achieves its download rate $d_i = d_{\mathrm{S},i} + \sum_{j\in\mathcal{N}_\mathrm{P}\setminus\{i\}} d_{j,i} = \sum_{i\in\mathcal{N}_\mathrm{P}} d_{\mathrm{S},i} = U_\mathrm{S}$. As a result, in this case, all the peers can finish file downloading at $t = F/U_\mathrm{S}$, which is also the time required for the server S sends all the portion of the file to the system. In other words, the bit-level flow like Fig. 2 is kept during file downloading for all peers, and thus the exchange of fresh contents between peers is guaranteed.

In other cases, the bit-level flow would be more complicated after the server S finishes sending all the pieces to the corresponding peers at $t = F/U_\mathrm{S}$. In the normal P2P file distribution [11], the exchange of fresh contents between peers can still be guaranteed by making the server S send the end of each piece j to peer i $(i,j \in \mathcal{N}_\mathrm{P}, i \neq j)$ as shown in Fig. 3. In the TFT-based P2P file distribution, more sophisticated piece sending of the server S will be required. In this paper, we assume that the exchange of fresh contents between peers is guaranteed during file downloading and the detailed analysis of the bit-level flow is one of the future directions. In other words, our approach will give at least the lower bound of the minimum distribution time of the TFT-based P2P file distribution.

Finally, Table 1 summarizes the notations used in the system model.

Table 1: Notations.

Notation	Definition
S	Server id
$\mathcal{N}_{\mathrm{P}}$	Set of peers $\mathcal{N}_{\mathrm{P}} = \{1, \ldots, N_{\mathrm{P}}\}$
$\mathcal{N}$	Set of nodes $\mathcal{N} = \{\mathrm{S}\} \cup \mathcal{N}_{\mathrm{P}},\ N = \lvert\mathcal{N}\rvert$
U_i	Upload capacity of node $i \in \mathcal{N}$
$d_{j,i}$	Download rate from node $j \in \mathcal{N}$ to leecher $i \in \mathcal{N}_{\mathrm{P}}$
d_i	Total download rate of peer $i \in \mathcal{N}_{\mathrm{P}}$
d	Total download rate of a peer
F	File size
T	Distribution time

4 Linear program for minimization of distribution time of TFT-based P2P file distribution

Considering the system model in Section 3, we formulate the minimization problem of distribution time of the TFT-based P2P file distribution as an LP. We first give the formulation for a general case (Section 4.1), and then provide that for a special case where peers are categorized into multiple classes according to their upload capacities (Section 4.2).

4.1 General case

In the general case, we can formulate the minimization problem of distribution time of the TFT-based P2P file distribution as the following LP.

$$
\begin{aligned}
\min\ & F/d, && && (2)\\
\text{s.t. } & d_i = d && \forall i \in \mathcal{N}_{\mathrm{P}}, && (3)\\
& d_i = d_{\mathrm{S},i} + \sum_{j \in \mathcal{N}_{\mathrm{P}} \setminus \{i\}} d_{j,i} && \forall i \in \mathcal{N}_{\mathrm{P}}, && (4)\\
& \sum_{j \in \mathcal{N}_{\mathrm{P}} \setminus \{i\}} d_{i,j} \le U_i && \forall i \in \mathcal{N}, && (5)\\
& d_{i,j} \le d_{\mathrm{S},i} && \forall i, j \in \mathcal{N}_{\mathrm{P}}, i \neq j, && (6)\\
& d_{i,j} = d_{j,i} && \forall i, j \in \mathcal{N}_{\mathrm{P}}, i < j. && (7)
\end{aligned}
$$

Objective function (2) means that the distribution time F/d should be minimized. Constraint (3) ensures that all leechers have the same download rate, and thus they will simultaneously finish the file download. Here, the download rate of each leecher $i \in \mathcal{N}_{\mathrm{P}}$ is given by the sum of download rates from the server S and other leechers, as in constraint (4). d, d_i ($i \in \mathcal{N}_{\mathrm{P}}$), and $d_{i,j}$ ($i \in \mathcal{N}, j \in \mathcal{N}_{\mathrm{P}}, i \neq j$) are the decision variables and the number of them is given by $1 + N_{\mathrm{P}} + N_{\mathrm{P}}(N_{\mathrm{P}} - 1) = O(N_{\mathrm{P}}^2)$.

Constraint (5) guarantees that the total upload rate of node $i \in \mathcal{N}$ should be equal or less than its upload capacity U_i. Since each leecher $i \in \mathcal{N}_{\mathrm{P}}$ can only forward the piece retrieved from the server S, constraint (6) ensures that its upload rate to other leecher $j \in \mathcal{N}_{\mathrm{P}} \setminus \{i\}$ (i.e., $d_{i,j}$) should be equal or less than its download rate from the server S (i.e., $d_{\mathrm{S},i}$). Note that each leecher i can copy and forward the piece retrieved from the server S to other leechers under its upload capacity constraint in (5). Finally, constraint (7) means the TFT constraint where leechers $i \in \mathcal{N}_{\mathrm{P}}$ and $j \in \mathcal{N}_{\mathrm{P}} \setminus \{i\}$ exchange the same amount of data. The number of above-mentioned constraints becomes $N_{\mathrm{P}} + N_{\mathrm{P}} + (N_{\mathrm{P}} + 1) + N_{\mathrm{P}}(N_{\mathrm{P}} - 1) + \binom{N_{\mathrm{P}}}{2} = O(N_{\mathrm{P}}^2)$.

All the variables are real and both the objective function and all the constraints are linear, and thus this optimization problem is an LP.

4.2 Special case: Multi-class upload capacity

The general formulation in Section 4.1 enables precise analysis under arbitrary capacity distribution at the cost of computation complexity depending on the system scale N_{P}. In the current Internet, the access link can be categorized into multiple classes (e.g., digital subscriber line (DSL), cable Internet (CATV), and fiber to the home (FTTH)). In such cases, the general formulation can be reduced to the following multi-class formulation where each peer belongs to one of C classes denoted by $\mathcal{C} = \{1, \ldots, C\}$.

$$\min \; F/d, \tag{8}$$

$$\text{s.t. } d_c = d, \qquad \forall c \in \mathcal{C}, \tag{9}$$

$$d_c = d_{\mathrm{S},c} + (\alpha_c N_{\mathrm{P}} - 1) d_{c,c} + \sum_{c' \in \mathcal{C} \setminus \{c\}} \alpha_{c'} N_{\mathrm{P}} d_{c',c} \quad \forall c \in \mathcal{C}, \tag{10}$$

$$(\alpha_c N_{\mathrm{P}} - 1) d_{c,c} + \sum_{c' \in \mathcal{C} \setminus \{c\}} \alpha_{c'} N_{\mathrm{P}} d_{c,c'} \leq U_c \qquad \forall c \in \mathcal{C}, \tag{11}$$

$$\sum_{c \in \mathcal{C}} \alpha_c N_{\mathrm{P}} d_{\mathrm{S},c} \leq U_{\mathrm{S}}, \tag{12}$$

$$d_{c,c'} \leq d_{\mathrm{S},c} \qquad \forall c, c' \in \mathcal{C}, \tag{13}$$

$$d_{c,c'} = d_{c',c} \qquad \forall c, c' \in \mathcal{C}, c < c'. \tag{14}$$

Here α_c (resp. U_c) denote the fraction (resp. upload capacity) of class-c peer ($0 \leq \alpha_c \leq 1, \sum_{c \in \mathcal{C}} \alpha_c = 1, U_c > 0$). The objective function is the same as that in the general case. Constraint (9) ensures that the download rate of each class should be identical. Constraint (10) gives the total download rate of class-c peer that consists of the download rate from the server S, those from other $(\alpha_c N_{\mathrm{P}} - 1)$ peers at the same class c, and those from other $\alpha_{c'} N_{\mathrm{P}}$ peers at different classes $c' \in \mathcal{C} \setminus \{c\}$. Since peers in the same class are homogeneous, we only need to manage the total download rate per class. In general, the number of classes, C, tends to be much smaller than the number of peers, N_{P}. Constraint (11) means that the total upload rate of class-c peer should be equal or less than its upload capacity U_c. Constraint (12) is the upload capacity constraint for the server S. Constraint (13) guarantees that each peer can provide piece retrieved from the server S to other peers. Finally, constraint (14) gives the TFT constraint.

The decision variables of the multi-class formulation are d, d_c, $d_{\mathrm{S},c}$, and $d_{c,c'}$ ($c, c' \in \mathcal{C}$). The number of them is $1+C+C^2 = O(C^2)$, which is much smaller than that in the general formulation, i.e., $O(N_{\mathrm{P}}^2)$. In addition, the number of constraints in the multi-class formulation becomes $C+C+C+1+C^2+\binom{C}{2} = O(C^2)$, which is also much smaller than that in the general formulation, i.e., $O(N_{\mathrm{P}}^2)$.

5 Numerical results

In this section, we reveal the fundamental characteristics of the TFT-based P2P file distribution by solving the LP formulated in Section 4 using CPLEX. We use the server with Intel Xeon E5-2650v4 (12 cores and 2.20 GHz) and 256 GB memory to obtain the following results.

5.1 Evaluation scenario

We consider a distribution system where one server S distributes a file of size 100 ($F = 100$). In the conventional *client–server model*, only the server S sends the file of size F to all the N_{P} peers at the same rate $U_{\mathrm{S}}/N_{\mathrm{P}}$, which makes the minimum distribution time $T_{\mathrm{min}}^{\mathrm{CS}}(N_{\mathrm{P}})$ as follows:

$$T_{\mathrm{min}}^{\mathrm{CS}}(N_{\mathrm{P}}) = \frac{F}{U_{\mathrm{S}}/N_{\mathrm{P}}} = \frac{FN_{\mathrm{P}}}{U_{\mathrm{S}}}. \tag{15}$$

Eq. (15) indicates that $T_{\mathrm{min}}^{\mathrm{CS}}(N_{\mathrm{P}})$ linearly increases with the system scale, N_{P}, and is not scalable.

On the other hand, if all the N_{P} peers are altruistic (cooperative) to upload retrieved pieces to others (i.e., *altruistic model*), the minimum distribution time $T_{\mathrm{min}}^{\mathrm{ALT}}(N_{\mathrm{P}})$ is given as follows [11]:

$$T_{\mathrm{min}}^{\mathrm{ALT}}(N_{\mathrm{P}}) = \frac{F}{\min\left(N_{\mathrm{P}}^{-1}(U_{\mathrm{S}} + \sum_{i\in\mathcal{N}_{\mathrm{P}}} U_i), U_{\mathrm{S}}\right)}. \tag{16}$$

If $N_{\mathrm{P}} \to \infty$, Eq. (16) approaches

$$\lim_{N_{\mathrm{P}}\to\infty} T_{\mathrm{min}}^{\mathrm{ALT}}(N_{\mathrm{P}}) = \frac{F}{\min(\overline{U}_{\mathrm{P}}, U_{\mathrm{S}})}, \tag{17}$$

where $\overline{U}_{\mathrm{P}} = N_{\mathrm{P}}^{-1}\sum_{i\in\mathcal{N}_{\mathrm{P}}} U_i$. Eq. (17) indicates that $\lim_{N_{\mathrm{P}}\to\infty} T_{\mathrm{min}}^{\mathrm{ALT}}(N_{\mathrm{P}})$ is independent of N_{P}, and thus much scalable than the client–server model. Note that the altruistic model is an ideal case where no peers are willing to be free riders. In practical systems, there may be $\beta_{\mathrm{F}}N_{\mathrm{P}}$ ($0 \le \beta_{\mathrm{F}} \le 1$) free riders in case of the P2P file distribution systems without the TFT constraint (i.e., *no*

Table 2: Default parameter settings.

Parameter	Default value
Number of peers, N_P	1000
Upload capacity of server S, U_S	100
Upload capacity of high-speed peer, U_H	2
Upload capacity of low-speed peer, U_L	1
Fraction of high-speed peers, α_H	0.2
Fraction of free riders, β_F	0
File size, F	100

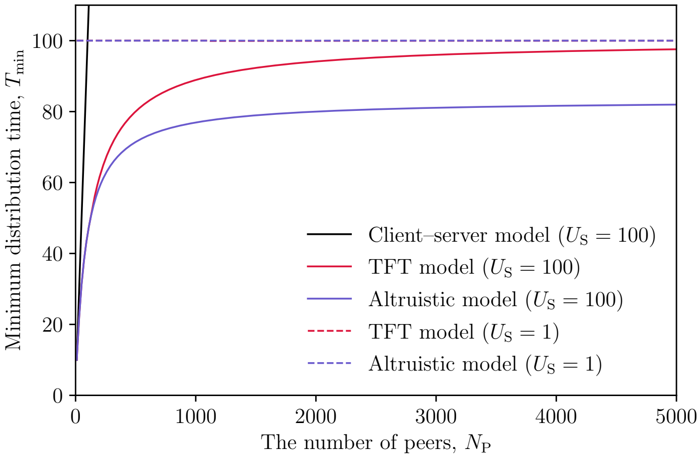


Fig. 4: Scalability of file distribution to the number of peers ($U_\mathrm{H} = 2, U_\mathrm{L} = 1, \alpha_\mathrm{H} = 0.2, F = 100$).

TFT model). The altruistic model can be regarded as a special class of the no TFT model with $\beta_\mathrm{F} = 0$.

In this paper, we focus on the TFT-based P2P file distribution (i.e., *TFT model*), which can achieve $\beta_\mathrm{F} = 0$ by forcing each pair of leechers to exchange the same amount of data under the TFT constraint. $T^\mathrm{CS}_\mathrm{min}(N_\mathrm{P})$ (resp. $T^\mathrm{ALT}_\mathrm{min}(N_\mathrm{P})$) can be regarded as the upper (resp. lower) bound of the minimum distribution time $T^\mathrm{TFT}_\mathrm{min}(N_\mathrm{P})$ of the TFT model.

In what follows, to reveal the fundamental characteristics of the TFT-based P2P file distribution, we mainly focus on the two-class upload capacity case where the system consists of high-speed peers and low-speed peers ($\mathcal{C} = \{\mathrm{H}, \mathrm{L}\}$). Table 2 gives the default parameter settings.

5.2 Scalability of TFT-based P2P file distribution

We first focus on the scalability of the TFT-based P2P file distribution to the number of peers, N_P. Fig. 4 illustrates the relationship between N_P and

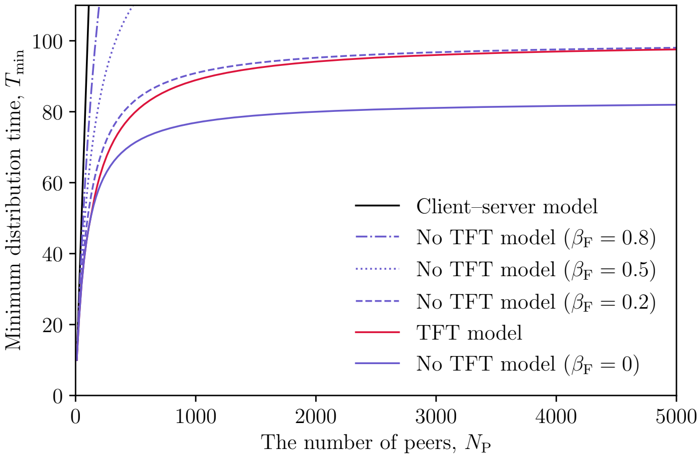


Fig. 5: Impact of fraction of free riders ($U_{\mathrm{S}} = 100, U_{\mathrm{H}} = 2, U_{\mathrm{L}} = 1, \alpha_{\mathrm{H}} = 0.2, F = 100$).

the minimum distribution time $T_{\min}$ of three models (i.e., client–server model, altruistic model, and TFT model) when the upload capacity of the server S, U_{S}, is set to be 100 or 1.

First, we confirm that the minimum distribution time of the client–server model, $T_{\min}^{\mathrm{CS}}(N_{\mathrm{P}})$, linearly increases with N_{P}, as in Eq. (15). On the other hand, we observe that the two P2P models (i.e., altruistic model and TFT model) can achieve much smaller minimum distribution time, regardless of U_{S}. Next, we focus on the case with $U_{\mathrm{S}} = 100$. The minimum distribution time of the altruistic model, $T_{\min}^{\mathrm{ALT}}(N_{\mathrm{P}})$, approaches $F/\min(\overline{U}_{\mathrm{P}}, U_{\mathrm{S}}) = F/\overline{U}_{\mathrm{P}}$ with increase of N_{P}. We observe that the minimum distribution time of the TFT model, $T_{\min}^{\mathrm{TFT}}(N_{\mathrm{P}})$, first shows the similar tendency (i.e., $N_{\mathrm{P}} \leq 120$) and then gradually becomes larger than $T_{\min}^{\mathrm{ALT}}(N_{\mathrm{P}})$. Note that we can confirm that $T_{\min}^{\mathrm{TFT}}(N_{\mathrm{P}})$ is much smaller (scalable) than $T_{\min}^{\mathrm{CS}}(N_{\mathrm{P}})$. The similar tendency of the distribution time of the altruistic model has also been observed in the existing studies [11,17,23,27]. As mentioned in Section 5.1, the altruistic model is equivalent with the no TFT model with $\beta_{\mathrm{F}} = 0$. The impact of the fraction of free riders, β_{F}, will be discussed in Section 5.3.

Finally, we focus on the case with $U_{\mathrm{S}} = 1$. The distribution time of the altruistic model, $T_{\min}^{\mathrm{ALT}}(N_{\mathrm{P}})$, is independent of N_{P} and becomes F/U_{S} as in Eq. (16). In this case, we also confirm that the TFT model achieves the same performance as the altruistic model.

5.3 Impact of fraction of free riders

In Section 5.2, we confirmed that the altruistic model outperforms the TFT model under the ideal situations where all the peers are cooperative. In practi-

Submitted version

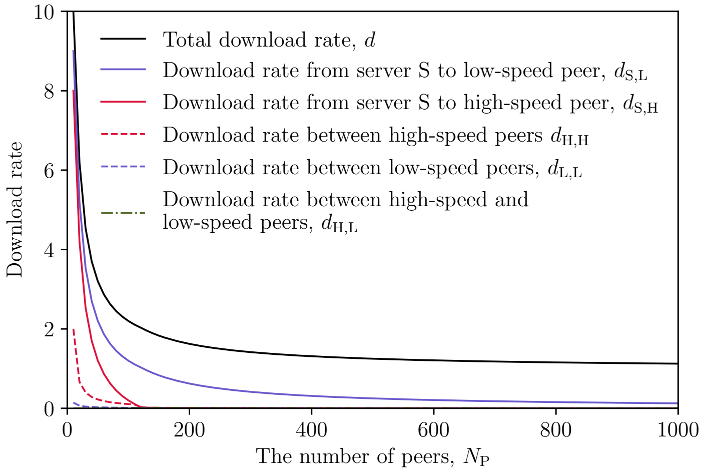


Fig. 6: Optimal download rate between nodes (TFT model, $U_{\mathrm{S}} = 100, U_{\mathrm{H}} = 2, U_{\mathrm{L}} = 1, \alpha_{\mathrm{H}} = 0.2, F = 100$).

cal, peers are potentially selfish and may be free riders [3]. Since the low-speed peers tends to deplete their own upload capacities, we select $\beta_{\mathrm{F}} N_{\mathrm{P}}$ free riders from all N_{P} nodes in ascending order of their upload capacities. Note that the distribution time of the no TFT model, $T_{\min}^{\mathrm{NOTFT}}(N_{\mathrm{P}})$, under the free-riding scenario can be derived by regarding the free riders as peers without upload capacities (i.e., $U_i = 0$).

Fig. 5 presents the impact of the fraction of free riders, β_{F}, on the distribution time of the no TFT model, $T_{\min}^{\mathrm{NOTFT}}(N_{\mathrm{P}})$. For comparison purpose, we also show the distribution time of the client-server model and that of the TFT model. We observe that $T_{\min}^{\mathrm{NOTFT}}(N_{\mathrm{P}})$ increases with β_{F}. Please note that $T_{\min}^{\mathrm{NOTFT}}(N_{\mathrm{P}})$ with $\beta_{\mathrm{F}} = 0$ (resp. $T_{\min}^{\mathrm{NOTFT}}(N_{\mathrm{P}})$ with $\beta_{\mathrm{F}} = 1$) is equivalent to $T_{\min}^{\mathrm{ALT}}(N_{\mathrm{P}})$ (resp. $T_{\min}^{\mathrm{CS}}(N_{\mathrm{P}})$). We confirm that $T_{\min}^{\mathrm{NOTFT}}(N_{\mathrm{P}})$ with $\beta_{\mathrm{F}} = 0.2$ is similar to $T_{\min}^{\mathrm{TFT}}(N_{\mathrm{P}})$. Fig. 5 implies that the TFT-based P2P file distribution will perform better than the normal P2P file distribution under practical situations where most peers are prone to be free riders.

In what follows, we further reveal the potential of the TFT-based P2P file distribution, and thus focus on the two models (i.e., TFT model and altruistic model).

5.4 Optimal download rate between nodes

In this section, we investigate the optimal download rate between nodes of the TFT model. Fig. 6 depicts the relationship between the number of peers, N_{P}, and each download rate between nodes (i.e., $d, d_{\mathrm{S,L}}, d_{\mathrm{S,H}}, d_{\mathrm{H,H}}, d_{\mathrm{L,L}}$, and $d_{\mathrm{L,H}}$). We first observe that the server S's upload rate to one low-speed peer,

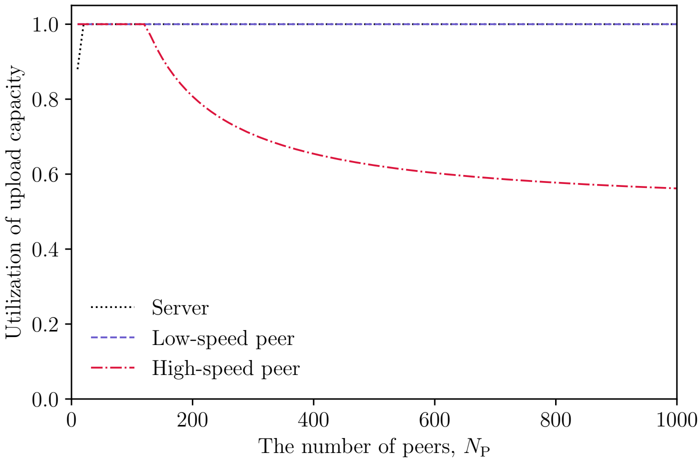


Fig. 7: Utilization of upload capacities of nodes (TFT model, $U_{\mathrm{S}} = 100, U_{\mathrm{H}} = 2, U_{\mathrm{L}} = 1, \alpha_{\mathrm{H}} = 0.2, F = 100$).

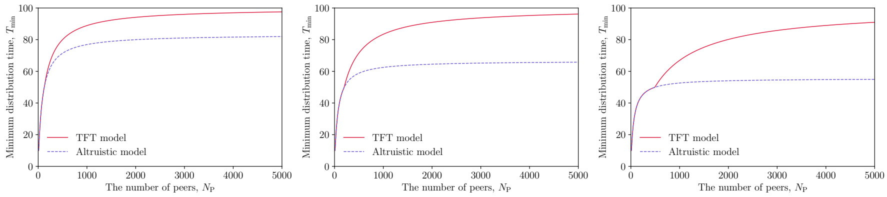
$d_{\mathrm{S,L}}$, is always larger than that to one high-speed peer, $d_{\mathrm{S,H}}$. This result is consistent with the ILP-based analytical result [6] and the most deprived peer selection [23].

Next, focusing on the download rate between peers, we confirm that the download rate between high-speed peers, $d_{\mathrm{H,H}}$, is larger than that between low-speed peers, $d_{\mathrm{L,L}}$, and that between high-speed and low-speed peers, $d_{\mathrm{H,L}}$, in case of $N_{\mathrm{P}} \leq 120$. In this range, both the server's support for low-speed peers and the abundant data exchange between high-speed peers enable the TFT model to achieve the same performance as the altruistic model as shown in Fig. 4. However, we also confirm that further increase of N_{P} tends to lack of download rate from the server S at both high-speed and low-speed peers, which results in the deterioration of the minimum distribution time of the TFT model compared to that of the altruistic model.

Finally, we also observe that the download rate between high-speed and low-speed peers, $d_{\mathrm{H,L}}$, is almost zero, regardless of N_{P}. This optimal results are similar to the bandwidth clustering phenomena observed in the BitTorrent systems, which not only adopt the TFT strategy but also other complicated mechanisms like optimistic unchoking [20, 28].

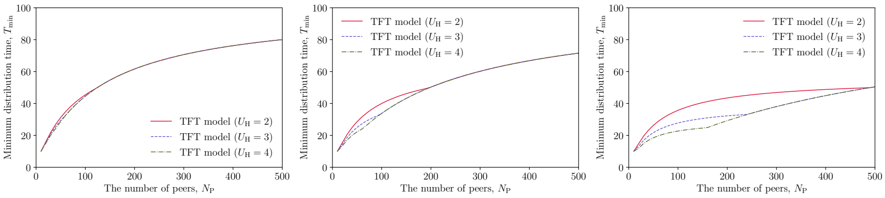
5.5 Impact of fraction of high-speed peers

In case of the altruistic model, increase of high-speed peers (i.e., $\alpha_{\mathrm{H}} N_{\mathrm{P}}$) will improve the distribution time thanks to their cooperative assist with abundant upload capacities. On the contrary, in case of the TFT model, the contribution of such high-speed peers may be suppressed due to the TFT constraint.



(a) $\alpha_H = 0.2$. (b) $\alpha_H = 0.5$. (c) $\alpha_H = 0.8$.

Fig. 8: Impact of the fraction of high-speed peers on distribution time ($U_S = 100, U_H = 2, U_L = 1, F = 100$).



(a) $\alpha_H = 0.2$. (b) $\alpha_H = 0.5$. (c) $\alpha_H = 0.8$.

Fig. 9: Impact of capacity difference between high-speed and low-speed peers on distribution time ($U_S = 100, U_L = 1, F = 100$).

Fig. 8 presents the impact of the fraction of the high-speed peers, α_H, on the distribution time in case of the altruistic model and TFT model. We first note here that the distribution time of both models becomes identical in case of $\alpha_H = 0$, according to the analytical result in [11]. When α_H increases, the increase of high-speed peers contributes to reduction of the distribution time in case of the both models. When the system scale is relatively small (e.g., $N_P \leq 130, 190$, and 490 in case of $\alpha_H = 0.2, 0.5$, and 0.8), the TFT model can achieve the same performance as the altruistic model. When N_P further increases, we also confirm that the improvement degree of the altruistic model becomes large than that of the TFT model.

In what follows, we further investigate which situations the TFT model becomes attractive.

5.6 Impact of capacity difference between high-speed and low-speed peers

Fig. 9 illustrates the impact of the capacity difference between high-speed and low-speed peers on the distribution time of the TFT model when the upload capacity of high-speed peer, U_H, is set to be $2, 3$, and 4, and the fraction of high-speed peers, α_H is set to be $0.2, 0.5$, and 0.8, respectively. When α_H is large (i.e., $\alpha_H = 0.8$), we observe that increase of U_H contributes to reduction of the distribution time thanks to the abundant data exchange among high-speed peers. However, we also confirm that increase of N_P leads to the lack of

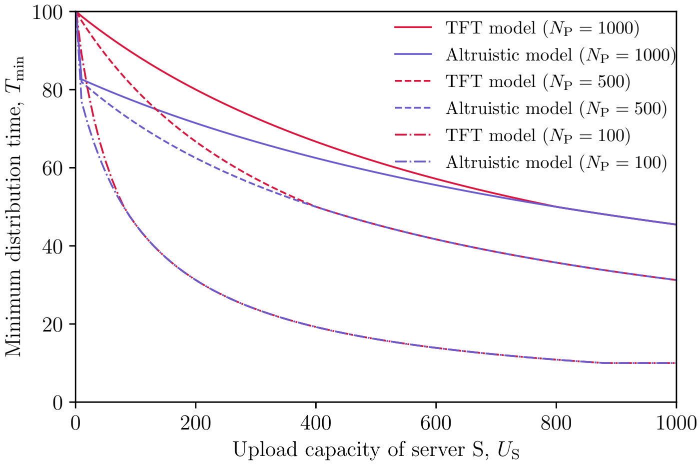


Fig. 10: Impact of upload capacity of server on distribution time ($U_{\mathrm{H}} = 2, U_{\mathrm{L}} = 1, \alpha_{\mathrm{H}} = 0.2, F = 100$).

the server S's upload capacity, which results in no performance improvement with the increase of U_{H}.

5.7 Impact of upload capacity of server

In this section, we further investigate the impact of the server S' upload capacity, U_{S}. Fig. 10 depicts the relationship between U_{S} and the distribution time when the number of peers, N_{P}, is set to be $100, 500$, and 1000, respectively. We first observe that the distribution time of the TFT model becomes 19.6–22% worse than that of the altruistic model if U_{S} is small. On the other hand, increase of U_{S} contributes to improving the performance of the TFT model and makes it competitive with that of the altruistic model when U_{S} is equal or greater than $80, 370$, and 730 in case of $N_{\mathrm{P}} = 10, 500$, and 1000, respectively.

5.8 Impact of network partitioning

So far we assume that all the peers can communicate with other peers. In other words, we regard the P2P network as a full-mesh network. In actual systems, it is desirable for each peer to know part of the whole peers to reduce the control and communication overhead. In this section, focusing on two kinds of network structures: (1) the number of peers that each peer can communicate and (2) connectivity among peers, we reveal how these structures affect the optimality of the TFT-based P2P file distribution. For this purpose, we first partition the whole system of N_{P} peers into K ($K \geq 1$) disjoint sub-systems, each of which has the same number of peers, $N_{\mathrm{P}}^{K} = N_{\mathrm{P}}/K$, and has the same upload

Table 3: Impact of Network Partitioning ($N_{\mathrm{P}} = 1280, U_{\mathrm{S}} = 128, \alpha_{\mathrm{H}} = 0.2$).

K	1	2	4	8	16	32	64
N_{P}^K	1280	640	320	160	80	40	20
U_{S}	128	64	32	16	8	4	2
$T_{\min}^{\mathrm{TFT}}(N_{\mathrm{P}}^K)$	88.9	88.9	89.0	89.0	89.2	89.4	90.0

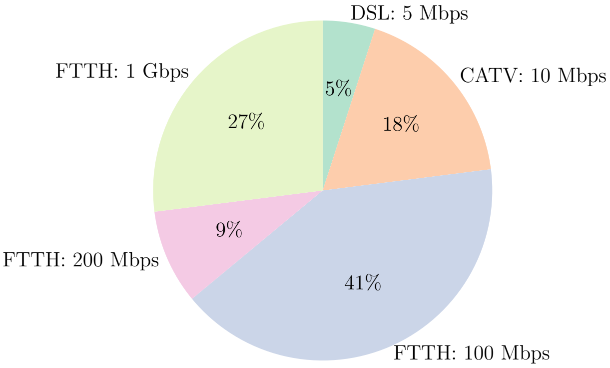


Fig. 11: Practical upload capacity distribution.

capacity of the server S, $U_{\mathrm{S}}^K = U_{\mathrm{S}}/K$. By comparing the system performance of the original (whole) system with that of the sub-systems, we can reveal how the network partitioning affects the optimality.

Table 3 presents the relationship between the number of sub-systems, K, and the distribution time of the TFT model, $T_{\min}^{\mathrm{TFT}}(N_{\mathrm{P}}^K)$, when we set K to be 2^k ($k = 0, 1, \ldots, 6$) and set N_{P} and U_{S} to be 1280 and 128, respectively. Since peers in a certain sub-system cannot communicate with those in other sub-systems, network partitioning has a possibility to degrade the system performance. However, we confirm that the degree of such degradation is limited in the range of 0.019–0.621%. The similar results were also observed in the ILP-based analysis for small-scale systems [8]. This results indicate that the whole system can be partitioned into multiple small sub-systems while keeping the optimality of distribution time, which will contribute to realizing the optimal P2P file distribution in an autonomous and distributed manner.

5.9 Evaluation under practical upload capacity distribution

Finally, we evaluate the effectiveness of the TFT-based P2P file distribution under more practical upload capacity distribution. According to the number and distribution of fixed-line broadband service subscriptions in Japan [29, Chap. 5], we prepare the fixed-line access scenario where there are 39,350,000 subscriptions (N_{P} = 39,350,000) and five-class access links: DSL (5 Mbps), CATV (10 Mbps), FTTH (100 Mbps), FTTH (200 Mbps), and FTTH (1 Gbps).

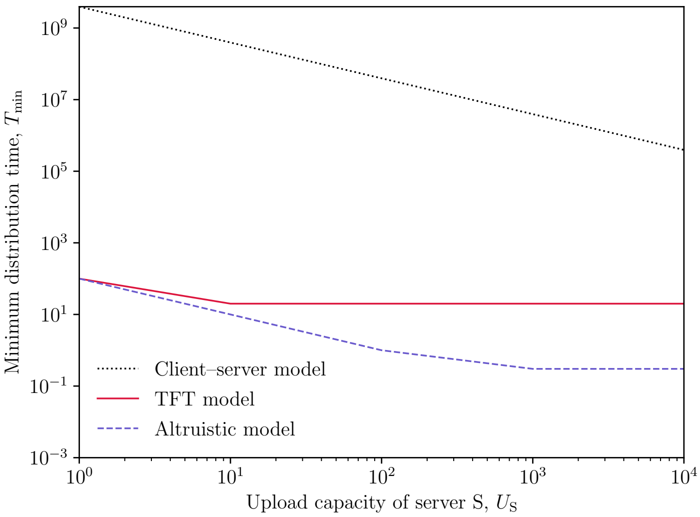


Fig. 12: Minimum distribution time under practical upload capacity distribution ($N_{\mathrm{P}} = 39{,}350{,}000, F = 100$).

Fig. 11 presents the upload capacity distribution of the fixed-line access scenario.

Fig. 12 illustrates the relationship between the upload capacity of server S and the minimum distribution time for the three models (i.e., client–server model, TFT model, and altruistic model) when $F = 100$ [Mbits]. We can confirm the similar tendency observed in the above-mentioned results for the two-class scenario. First, we confirm that the two P2P models (i.e., TFT model and altruistic model) are still much scalable than the client–server model. Next, we focus on the two P2P models. When U_{S} is small, the upload capacity of the server S tends to be bottleneck and thus the TFT model achieves almost the same performance as the altruistic model. When U_{S} becomes large, the TFT model cannot achieve the same performance as the altruistic model, due to the TFT constraints with low-speed peers. However, we also confirm that the performance difference between them is limited and U_{S} can be much small compared with the system scale, N_{P}, to achieve speedy file distribution.

6 Conclusions

In this paper, with the help of the fluid model, we have developed an LP to minimize the distribution time of the TFT-based P2P file distribution under the assumption that all the peers simultaneously finish file downloading. By considering the fact that access link can be categorized into multiple classes in the current Internet, we have further extended the general formulation to a multi-class formulation whose problem size does not depend on the system scale. Through numerical results, we have revealed the following characteristics of the TFT-based P2P file distribution compared with the conventional client-server file distribution and normal P2P file distribution.

1. The TFT-based P2P file distribution (TFT model) has slightly larger minimum distribution time than the ideal P2P file distribution with altruistic/cooperative peers (altruistic model) but it is still much scalable than the conventional client–server file distribution.
2. The TFT model can achieve shorter minimum distribution time than the normal P2P file distribution (no TFT model) when more than 20% peers become free riders.
3. If the server's upload capacity is bottleneck, increase of high-speed peers and capacity difference between high-speed peers and low-speed peers have limited impact on the performance of the TFT model.
4. If the server's upload capacity is abundant, piece exchange among peers is intensified, and thus the TFT model can achieve almost the same performance as the altruistic model.
5. The P2P network can be partitioned into multiple small-scale sub systems while keeping the minimum file distribution time, which will contribute to achieving autonomous and distributed control.
6. The TFT-based P2P file distribution is much scalable than the client–server file distribution even under the practical scenario based on the number and distribution of fixed-line broadband service subscriptions in Japan.

In future work, we plan to analyze the bit-level flow to achieve the minimum file distribution time of the TFT-based P2P file distribution.

Acknowledgements The author thanks Mr. Yohei Nishi for his valuable comments.

Conflict of interest

The author declares that he has no conflict of interest.

References

1. Carbunaru, C., Teo, Y.M., Leong, B., Ho, T.: Modeling Flash Crowd Performance in Peer-to-Peer File Distribution. IEEE Transactions on Parallel and Distributed Systems **25**(10), 2617–2626 (2014). DOI 10.1109/TPDS.2013.220
2. Deploy and Update Windows 10. available at `https://docs.microsoft.com/en-us/windows/deployment/`
3. Hughes, D., Coulson, G., Walkerdine, J.: Free Riding on Gnutella Revisited: The Bell Tolls? IEEE Distributed Systems Online **6**(6), 1–13 (2005). DOI 10.1109/MDSO.2005.31
4. BitTorrent. available at `https://www.bittorrent.com`
5. Azzedin, F., Yahaya, M.: Modeling BitTorrent Choking Algorithm Using Game Theory. Future Generation Computer Systems **55**, 255–265 (2016). DOI 10.1016/j.future.2015.02.007
6. Hasegawa, M., Sasabe, M., Takine, T.: Analysis of Optimal Scheduling in Tit-for-Tat-Based P2P File Distribution. IEICE Transactions on Communications **97**(12), 2650–2657 (2014)

7. Sasabe, M.: Analysis of Optimal Piece Flow in Tit-for-Tat-Based P2P Streaming. Computer Networks **139**(7), 60–69 (2018)
8. Sasabe, M.: Topological Influence on Optimality of Tit-for-Tat Based P2P Content Distribution. Peer-to-Peer Networking and Applications **13**(1), 243–254 (2020). DOI 10.1007/s12083-019-00763-x
9. Nishi, Y., Sasabe, M., Kasahara, S.: Optimality Analysis of Locality-Aware Tit-for-Tat-Based P2P File Distribution. Peer-to-Peer Networking and Applications pp. 1–16 (2020). DOI 10.1007/s12083-020-00925-2
10. IBM ILOG CPLEX Optimizer. available at `https://www.ibm.com/analytics/cplex-optimizer`
11. Kumar, R., Ross, K.W.: Peer-Assisted File Distribution: The Minimum Distribution Time. In: Proc. of 1st IEEE Workshop on Hot Topics in Web Systems and Technologies, pp. 1–11 (2006). DOI 10.1109/HOTWEB.2006.355259
12. Chen, D., Batson, R.G., Dang, Y.: Applied Integer Programming. Wiley (2010)
13. Pouwelse, J., Garbacki, P., Epema, D., Sips, H.: The Bittorrent P2P File-Sharing System: Measurements and Analysis. In: Proc. of the 4th International Conference on Peer-to-Peer Systems, pp. 205–216 (2005)
14. Stutzbach, D., Rejaie, R.: Understanding Churn in Peer-to-Peer Networks. In: Proc. of the 6th ACM SIGCOMM Conference on Internet Measurement, pp. 189–202 (2006). DOI 10.1145/1177080.1177105
15. Ge, Z., Figueiredo, D., Jaiswal, S., Kurose, J., Towsley, D.: Modeling Peer-Peer File Sharing Systems. In: Proc. of IEEE INFOCOM, vol. 3, pp. 2188–2198 (2003). DOI 10.1109/INFCOM.2003.1209239
16. Tewari, S., Kleinrock, L.: On Fairness, Optimal Download Performance and Proportional Replication in Peer-to-Peer Networks. In: Proc. of the 4th IFIP-TC6 International Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems, pp. 709–717 (2005). DOI 10.1007/11422778_57
17. Ferragut, A., Paganini, F.: Queueing Analysis of Peer-to-Peer Swarms: Stationary Distributions and Their Scaling Limits. Performance Evaluation **93**, 47–62 (2015). DOI 10.1016/j.peva.2015.08.003
18. Qiu, D., Srikant, R.: Modeling and Performance Analysis of BitTorrent-like Peer-to-Peer Networks. ACM SIGCOMM Computer Communication Review **34**(4), 367–378 (2004). DOI 10.1145/1030194.1015508
19. Gaeta, R., Gribaudo, M., Manini, D., Sereno, M.: Analysis of Resource Transfers in Peer-to-Peer File Sharing Applications Using Fluid Models. Performance Evaluation **63**(3), 149–174 (2006). DOI 10.1016/j.peva.2005.01.001
20. Meulpolder, M., Pouwelse, J.A., Epema, D.H.J., Sips, H.J.: Modeling and Analysis of Bandwidth-Inhomogeneous Swarms in BitTorrent. In: Proc. of IEEE P2P, pp. 232–241 (2009). DOI 10.1109/P2P.2009.5284523
21. Murai, F., Rocha, A.A.d.A., Figueiredo, D.R., de Souza e Silva, E.A.: Heterogeneous Download Times in a Homogeneous BitTorrent Swarm. Computer Networks **56**(7), 1983–2000 (2012). DOI 10.1016/j.comnet.2012.02.009
22. Ferragut, A., Paganini, F.: Fluid Models of Population and Download Progress in P2P Networks. IEEE Transactions on Control of Network Systems **3**(1), 34–45 (2016). DOI 10.1109/TCNS.2015.2434092
23. de Souza e Silva, E., Leão, R.M.M., Menasché, D.S., Towsley, D.: On the Scalability of P2P Swarming Systems. Computer Networks **151**, 93–113 (2019). DOI 10.1016/j.comnet.2019.01.006
24. DeFigueiredo, D., Venkatachalam, B., Wu, S.F.: Bounds on the Performance of P2P Networks Using Tit-for-Tat Strategies. In: Proc. of IEEE P2P, pp. 11–18 (2007). DOI 10.1109/P2P.2007.32
25. Carbunaru, C., Teo, Y.M., Leong, B.: A Performance Study of Peer-Assisted File Distribution with Heterogeneous Swarms. In: Proc of IEEE 36th Conference on Local Computer Networks, pp. 341–349 (2011). DOI 10.1109/LCN.2011.6115314
26. Carbunaru, C., Teo, Y.M., Leong, B., Ho, T.: Modeling Flash Crowd Performance in Peer-to-Peer File Distribution. IEEE Transactions on Parallel and Distributed Systems **25**(10), 2617–2626 (2014). DOI 10.1109/TPDS.2013.220

27. Mendes, D.X., de Souza e Silva, E., Menasché, D., Leão, R., Towsley, D.: An Experimental Reality Check on the Scaling Laws of Swarming Systems. In: Proc. of IEEE INFOCOM, pp. 1–9 (2017). DOI 10.1109/INFOCOM.2017.8057126
28. Legout, A., Liogkas, N., Kohler, E., Zhang, L.: Clustering and Sharing Incentives in BitTorrent Systems. ACM SIGMETRICS Performance Evaluation Review **35**(1), 301–312 (2007). DOI 10.1145/1269899.1254919
29. Ministry of Internal Affairs and Communications, Japan: Information and Communications in Japan. https://www.soumu.go.jp/johotsusintokei/whitepaper/eng/WP2018/2018-index.html (2018)