Submitted version

Impact of Locality-awareness on Tit-for-Tat-based P2P File Distribution

Yohei Nishi, Masahiro Sasabe, and Shoji Kasahara
Graduate School of Science and Technology, Nara Institute of Science and Technology,
8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan.
nishi.yohei.no0@is.naist.jp, {m-sasabe, kasahara}@ieee.org

***Abstract*—Periodic update of software is one of preventive measures against malicious attacks. When the software is used by many users, e.g., Operating System (OS), the distribution server for update tends to be a bottleneck. To tackle this problem, several systems, e.g., Windows update, recently apply Peer-to-Peer (P2P) file distribution where clients called peers upload retrieved fragments of the whole content, i.e., pieces, to other peers. However, some peers will not be willing to upload pieces to others, which are called free riders, due to communication overhead. Tit-for-Tat (TFT) strategy in game theory can alleviate such free riding behavior by encouraging equivalent exchange of pieces among each pair of peers. In recent years, the optimality of P2P file distribution under strict TFT constraint has been analyzed. In this paper, considering the fact that the communication overhead inside a group, e.g., LAN or AS, is much less than that between different groups, we consider the locality-aware TFT-based P2P file distribution where the TFT constraint is relaxed for intra-group communication. We find that the minimization of average file download time in the relaxed TFT-based P2P file distribution can also be modeled as Integer Linear Programming (ILP), as in the existing work. Through numerical results, we show that the relaxed TFT model contributes to shortening the average file download time among peers by about 18.3% compared with the strict TFT-based model.**

***Index Terms*—P2P file distribution, Tit-for-Tat strategy, optimal piece flow, Integer Liner Programming (ILP), locality-aware.**

I. INTRODUCTION

Nowadays, periodic software update is essential to maintain the system secure against malicious attacks. When the software is used by many users, e.g., Operating System (OS), and/or new update files, e.g., security patch files, are released, users' accesses concentrate on the distribution server, which causes flash crowd [1] and makes the server a bottleneck in terms of CPU and network resources. To tackle the scalability issue, several systems, e.g., Windows update, have been applying Peer-to-Peer (P2P) technologies to assist the distribution by end-user devices [2]. In the P2P file distribution, a file is divided into multiple fragments called pieces. Clients called peers can download pieces from the server as well as from other peers. If the peers cooperatively exchange pieces with each other, the server can achieve the file distribution that can adapt to the dynamical change of user demands without investments in equipment.

However, peers are user devices and thus their behavior depends on the corresponding users' decision making. In particular, sending pieces to others consume their own upload capacity, which may degrade the quality of other network services running on their devices. As a result, it has been pointed out that many peers tend to become free riders that hesitate to upload pieces to others [3]. Since the increase of free riders decreases the total upload capacity of the system, incentive mechanisms are essential to promote the mutual cooperation among peers.

To alleviate the free-riding problem, several P2P file distribution systems, e.g., BitTorrent [4], applies the Tit-for-Tat (TFT) strategy in game theory. Simply speaking, the TFT strategy in the P2P file distribution can be regarded as the equivalent piece exchange where each peer has to provide pieces with others so as to retrieve its demanding pieces from them. Although the TFT constraint encourages the mutual cooperation among peers, it may also degrade the distribution performance, due to the equivalent piece exchanges between peers.

In recent years, the optimality of the TFT-based P2P content distribution and its optimal piece flow have been analyzed [5]–[7]. The piece flow is the information that describes the transfer timing of each piece from a node (i.e., server or peer) to a peer. Hasegawa et al. first formulated the piece-flow decision problem to minimize the average file download time as Integer Linear Programming (ILP) [5]. In [6], Sasabe found that the piece-flow decision problem for minimizing the average play-out delay in the TFT-based P2P streaming could also be modeled as ILP. The impact of the P2P topology on the optimality of the TFT-based P2P contribution was also analyzed [7]. These existing work obtained the optimal solutions by solving their problems using the existing solver, i.e., CPLEX [8], and revealed the optimal behavior (strategy) of the server and peers that yields the optimal flow by analyzing the optimal solutions.

However, these existing work did not consider the physical proximity among peers, e.g., LAN and Autonomous System (AS), which would affect the cooperative relationship among peers. In the following, we refer to such physically close peers as a *group*. As mentioned above, the free-riding behavior tends to come from avoiding the bandwidth consumption caused by uploading pieces. In case of the intra-group communication, each peer can easily send pieces to other group members with abundant network bandwidth, and thus the cost of uploading is negligible. On the other hand, in case of the inter-group communication, peers potentially tend to be free riders, due to the limited upload capacity of their access links.

Moreover, the traffic between peers that belong to different ASs may impose the transit cost on the related ISPs. In general, the transit cost is imposed on the lower tier ISP when it use the transit link to the upper tier ISP and it is determined according to the q-percentile charging policy [9]. In the q-percentile charging policy, the traffic within a fixed period, e.g., 5 minutes, is periodically measured and sorted in ascending order. The transit cost is determined based on the $q\%$-th largest amount of traffic. Thus, it is important to reduce the peak traffic so as to suppress the transit cost. From the viewpoint of the P2P content distribution, the reduction of the transit cost will contribute to the ISP-friendly content distribution. In this paper, we focus on the locality-aware TFT-based P2P file distribution where the TFT constraint is relaxed for intra-group communication. After modeling it as a discrete-time system, we formulate the optimal piece flow determination problem as two-stage ILP. At the first stage optimization, we aim to minimize the average file download time among peers. Next, we further try to minimize the peak traffic between groups at the second stage optimization. Through numerical results obtained by solving the ILP, we reveal the fundamental characteristics of the locality-aware TFT-based P2P file distribution.

The rest of the paper is organized as follows. Section II gives related work. In Section III, we model the locality-aware TFT-based P2P file distribution and formulate the optimal piece flow determination problem as two-stage ILP. After demonstrating numerical results in Section IV, we give conclusions in Section V.

II. RELATED WORK

A. Performance analysis of P2P content distribution

There are many studies on the performance analysis of the P2P content distribution [1], [3], [5]–[7], [10], [11]. In [1], the authors formulated an analytical model to estimate the available bandwidth of the P2P file distribution systems under flash crowds. In [3], the authors proposed a game theory model to mitigate the impact of free riders on the BitTorrent performance. Silva et al. proposed models to calculate the system throughput of the P2P swarming systems under distinct policies for both peer and piece selection [10]. The comprehensive survey of BitTorrent performance was given in [11]. The optimality of the TFT-based P2P content distribution was studied from the viewpoint of the content types (i.e., file distribution case [5] and on-demand/live streaming case [6]) and the topological influence on the system performance [7].

B. Locality-aware P2P content distribution

Since a P2P network is constructed over the top of the underlying physical network, e.g., the Internet, the transmission rate and latency between two peers highly depend on their physical closeness. If neighboring peers over the P2P network are physically close, they tend to be able to enjoy high transmission rate with low latency as well as the transit cost between ISPs will also be reduced. Achieving such locality-aware P2P content distribution is one of the most important issues [12]–[15].

In [12], Blond et al. analyzed the impact of the number of connections between peers belonging to different ISPs to the download time and communication overhead. In [13], Magharei et al. proposed a scheduling scheme for P2P streaming to reduce inter-ISP traffic. Zhang et al. evaluated the P2P content distribution based on the ISP information from the view points of intra-ISP traffic, economic benefits, and user efficiency [14]. In [15], Costa et al. proposed a locality-aware BitTorrent protocol for enterprise networks composed of multiple LANs.

In addition to these research activities, Microsoft has been starting to apply the locality-aware P2P technologies to assist Windows update [2]. Windows update newly support two kinds of P2P delivery schemes in addition to the conventional client–server delivery scheme. In the first P2P delivery scheme, user PCs can send and receive update files with other PCs in the same sub network, e.g., LAN. The second P2P delivery scheme also allow the user PCs exchange update files among them without the limitation of physical closeness.

In this paper, we focus on the fact that peers can easily upload pieces to physically close peers with abundant network bandwidth while they consume the limited upload capacity of their access links to provide pieces with physically distant peers. This indicates that the TFT constraint is still important for physically distant peers but may not be needed for physically close peers.

III. MODELING LOCALITY-AWARE TFT-BASED P2P FILE DISTRIBUTION AND ILP FORMULATION OF DETERMINATION OF ITS OPTIMAL PIECE FLOW

In this section, we model the locality-aware TFT-based P2P file distribution and formulate the determination of its optimal piece flow as two-stage ILP. In [5], the determination of the optimal piece flow under the strict TFT-based P2P file distribution is formulated as ILP and we extend it such that we can consider the physical proximity between peers. The first stage is used to obtain the optimal piece flow to minimize the average file download time among peers. The second stage is used to obtain the optimal piece flow to minimize the inter-group traffic (i.e., the maximum number of exchanged pieces between groups).

A. Model

As in [5], the TFT-based P2P file distribution is modeled as a discrete-time system. In the system, there are a server, denoted by $\mathcal{N}_{\mathrm{D}} = \{0\}$, and N_P peers, denoted by $\mathcal{N}_{\mathrm{P}} = \{1, \ldots, N_{\mathrm{P}}\}$. In what follow, the server and peers are collectively called *nodes*. For simplicity, we define the set of nodes as $\mathcal{N} = \mathcal{N}_{\mathrm{D}} \cup \mathcal{N}_{\mathrm{P}}$ whose size is $N = |\mathcal{N}|$. Peers play two kinds of roles, *leechers* and *seeds*, depending on their file retrieval states. Leechers are peers under file downloading while seeds are those that finish file downloading. The content file consists of M fragments, i.e., pieces, denoted by

TABLE I: Notations in the model.

Notation	Definition
$\mathcal{N}_{\mathrm{D}}$	The set of server,$\{0\}$
$\mathcal{N}_{\mathrm{P}}$	The set of peers,$\{1,2,\dots,N_{\mathrm{P}}\}$
$\mathcal{N}$	The set of nodes, $\mathcal{N}=\mathcal{N}_{\mathrm{D}}\cup\mathcal{N}_{\mathrm{P}}$, $(N=\lvert\mathcal{N}\rvert)$
$\mathcal{M}$	The set of pieces, $\{1,2,\dots,M\}$
$G_{i,j}$	Group relationship between peer i and peer j (1: they are in the same group, 0: otherwise)
C_i^{in}	Upload capacity of peer i to group members
C_i^{out}	Upload capacity of peer i to non-group members
$x_{t,k,i,j}$	Decision variables of piece flow
$z_{t,k,i}$	Variables of piece possession (1: possession, 0: missing)
$y_{t,i}$	Variables of peer's role (1: leechers, 0: servers/seeds)
τ_i	File download time of peer i
$\eta_{\max}$	Peak traffic between groups

$\mathcal{M}=\{1,\dots,M\}$. Without loss of generality, we consider the piece size is identical and set to be one.

The two sets of time steps are defined as $\mathcal{T}=\{0,\dots,T\}$ and $\mathcal{T}^{+}=\{1,\dots,T\}$, respectively. T should be large enough to ensure that all peers complete file downloading. The derivation of T is discussed in [5].

Next, we focus on the physical proximity (i.e., group relationship) between nodes. Suppose that there are G groups. Each node belongs to an exact one group and the group relationship between node i and node j is given as input parameters:

$$G_{i,j}=\begin{cases}1, & \text{if } i \text{ and } j \text{ belong to the same group,}\\ 0, & \text{otherwise.}\end{cases} \tag{1}$$

Note that the server $i\in\mathcal{N}_{\mathrm{D}}$ is separated from peers, i.e., $G_{i,j}=0$ $(j\in\mathcal{N}_{\mathrm{P}})$. From the definition, $G_{i,j}=G_{j,i}$ $(i,j\in\mathcal{N})$. Each node $i\in\mathcal{N}$ can upload at most C_i^{out} pieces per unit time to peers in different groups, i.e., $\{j\in\mathcal{N}_{\mathrm{P}}\mid G_{i,j}=0\}$. For simplicity, we assume that the inter-group upload capacity of node i, C_i^{out}, is a natural and finite number. Considering the asymmetry between uplink and downlink channel speeds, e.g., ADSL and cable Internet, we assume that the inter-group upload capacity of each peer becomes the bottleneck compared with its inter-group download capacity. Since the intra-group communication, e.g., LAN, is much faster than the inter-group communication, we assume that the intra-group upload capacity of peer $i\in\mathcal{N}_{\mathrm{P}}$, denoted by C_i^{in}, is much larger than C_i^{out}.

The initial state of the system at $t=0$ can be represented by the piece possession of each node, $z_{0,k,i}$ $(k\in\mathcal{M},i,j\in\mathcal{N})$. For example, if the server only has the whole content, $z_{0,k,i}=1$ $(k\in\mathcal{M},i,j\in\mathcal{N}_{\mathrm{D}})$ and $z_{0,k,i}=0$ $(k\in\mathcal{M},i,j\in\mathcal{N}_{\mathrm{P}})$ should be hold. In the succeeding time steps $t\in T^{+}$, the piece flow among nodes determines the system dynamics, which is given by the decision variables $x_{t,k,i,j}$ $(t\in\mathcal{T},k\in\mathcal{M},i,j\in\mathcal{N})$:

$$x_{t,k,i,j}=\begin{cases}1, & \text{if node } i \text{ sends piece } k \text{ to node } j \text{ at time } t,\\ 0, & \text{otherwise.}\end{cases}$$

With the initial condition $z_{0,k,i}$ and the decision variables $x_{t,k,i,j}$, we can also derive two kinds of variables, $z_{t,k,i}$ $(t\in\mathcal{T}^{+},k\in\mathcal{M},i\in\mathcal{N})$ and $y_{t,i}$ $(t\in\mathcal{T},i\in\mathcal{N})$. $z_{t,k,i}$ denotes piece possession of node i:

$$z_{t,k,i}=\begin{cases}z_{t-1,k,i}+\sum_{j\in\mathcal{N}}x_{s,k,j,i}, & \text{if } i\in\mathcal{N}_{\mathrm{P}},\\ 1, & \text{otherwise.}\end{cases} \tag{2}$$

(2) represents that the server $i\in\mathcal{N}_{\mathrm{D}}$ always has all pieces. In the case of peers, $z_{t,k,i}=1$ if peer i has piece k at time t, and otherwise $z_{t,k,i}=0$. $y_{t,i}$ denotes the role of peer i:

$$y_{t,i}=\begin{cases}1-\prod_{k\in\mathcal{M}}z_{t,k,i}, & \text{if } i\in\mathcal{N}_{\mathrm{P}},\\ 0, & \text{otherwise.}\end{cases} \tag{3}$$

In case of the server $i\in\mathcal{N}_{\mathrm{D}}$, $y_{t,i}=0$ $(t\in\mathcal{T})$. On the other hand, if peer $i\in\mathcal{N}_{\mathrm{P}}$ is a leecher at time t $(t\in\mathcal{T})$, $y_{t,i}=1$, and otherwise, $y_{t,i}=0$.

The notations used in the model are summarized in Table I.

B. First stage: Minimization of average file download time

At the first stage, we focus on the main objective of the P2P file distribution, i.e., minimization of average file download time among peers. The corresponding optimization problem P_1 can be formulated as follows.

$$\min\ \overline{\tau}, \tag{4}$$

$$\text{s.t. } z_{0,k,i}=1,\quad \forall k\in\mathcal{M},\ \forall i\in\mathcal{N}_{\mathrm{D}}, \tag{5}$$

$$z_{0,k,i}=0,\quad \forall k\in\mathcal{M},\ \forall i\in\mathcal{N}_{\mathrm{P}}, \tag{6}$$

$$x_{t,k,i,j}=\{0,1\},\quad \forall t\in\mathcal{T}^{+},\ \forall i,j\in\mathcal{N},\ i\neq j, \tag{7}$$

$$x_{t,k,i,i}=0,\quad \forall t\in\mathcal{T}^{+},\ \forall k\in\mathcal{M},\ \forall i\in\mathcal{N}, \tag{8}$$

$$x_{t,k,i,j}\le z_{t-1,k,i},\quad \forall t\in\mathcal{T}^{+},\forall k\in\mathcal{M},\forall i,j\in\mathcal{N},i\neq j, \tag{9}$$

$$x_{t,k,i,j}\le 1-z_{t-1,k,j},\quad \forall t\in\mathcal{T}^{+},\ \forall k\in\mathcal{M},\ \forall i,j\in\mathcal{N},\ i\neq j, \tag{10}$$

$$\sum_{j\in\mathcal{N}}x_{t,k,i,j}\le 1,\quad \forall t\in\mathcal{T}^{+},\forall k\in\mathcal{M},\forall i\in\mathcal{N}, \tag{11}$$

$$x_{t,k,i,j}\le y_{t-1,i},\ \forall t\in\mathcal{T}^{+},\forall k\in\mathcal{M},\forall i,j\in\mathcal{N}_{\mathrm{P}}, \tag{12}$$

$$\tau_i\le\tau_j,\quad \forall i,j\in\mathcal{N}_{\mathrm{P}},\ C_i^{\mathrm{out}}\ge C_j^{\mathrm{out}} \tag{13}$$

$$\sum_{k\in\mathcal{M},j\in\mathcal{N}}(1-G_{i,j})x_{t,k,i,j}\le C_i^{\mathrm{out}},\forall t\in\mathcal{T}^{+},\forall i\in\mathcal{N} \tag{14}$$

$$\sum_{k\in\mathcal{M},j\in\mathcal{N}}G_{i,j}x_{t,k,i,j}\le C_i^{\mathrm{in}},\ \forall t\in\mathcal{T}^{+},\forall i\in\mathcal{N}_{\mathrm{P}}, \tag{15}$$

$$\sum_{k\in\mathcal{M}}(x_{t,k,i,j}-x_{t,k,j,i})\le MG_{i,j},\ \forall t\in\mathcal{T}^{+},\ \forall i,j\in\mathcal{N}_{\mathrm{P}}, \tag{16}$$

The objective function (4) indicates the minimization of the average file download time τ, which is given by

$$\overline{\tau}=N_{\mathrm{P}}^{-1}\sum_{i\in\mathcal{N}_{\mathrm{P}}}\tau_i$$

where τ_i denotes the file download time of peer i and is equivalent to the duration of being a leecher as follows:

$$\tau_i=\sum_{t\in\mathcal{T}}y_{t,i}\quad \forall i\in\mathcal{N}_{\mathrm{P}}$$

Constraints (5) through (13) are identical those in case of the strict TFT-based P2P file distribution [5]. (5) and (6) define the initial condition at $t = 0$: The server has all pieces while all peers have node piece. (7) gives the range of the decision variable $x_{t,k,i,j}$ and the self loop is prohibited by (8). (9) means the feasibility of sending piece k from node i to others while (10) indicates the need for peer j to retrieve piece k. As in (11), peer i retrieves piece k from at most one node. (12) presents the seed departure scenario where peers are selfish and leave the system just after the completion of file downloading. (13) defines the distribution policy that ensures the peers finish file downloading in descending order of their inter-group upload capacity.

(14) indicates that each node i can send at most C_i^{out} pieces to peers in different groups at each time. Similarly, (15) is the constraint for the intra-group communication. (16) indicates the TFT strategy with the proximity-based relaxation between peer i and peer j. The left hand side of (16) is the difference of the numbers of pieces exchanged between peer i and peer j at time t. The right hand side of (16) become zero if peers i and j belong to the different group, i.e., $G_{i,j} = 0$. In this case, peers i and j have to exchange the same number of pieces, that is the TFT constraint is satisfied. On the other hand, if peers i and j belong to the same group, i.e., $G_{i,j} = 1$, the right hand side of (16) becomes M. Since M is the number of pieces, (16) is always satisfied, that is the TFT constraint is relaxed according to the proximity between them. Note that (16) is applied to each pair of peers and the server altruistically provides pieces with peers.

We observe that all the variables are binary or integer and all the equations are linear except $y_{t,i}$ in (3). $y_{t,i}$ includes the nonlinear term of the product of binary variables, i.e., $z_{t,k,i}$, but it can be linearlized [5], [16]. As a result, the problem P_1 can be formulated as ILP.

C. Second stage: Minimization of peak traffic between groups

By solving problem P_1, there may be more than one optimal solutions, each of which minimizes the average file download time among peers. At the second stage, we further aim to minimize the peak traffic between groups, which will contribute to reduce the communication cost of both peers and their belonging groups, e.g., ISPs. The corresponding optimization problem P_2 can be formulated by modifying P_2 as follows. First, the objective function is replaced with

$$\min \quad \eta_{\max},$$

where $\eta_{\max}$ is the peak traffic that is the maximum number of pieces exchanged between groups per unit time, and thus it satisfies the following constraint:

$$\sum_{k\in\mathcal{M},i,j\in\mathcal{N}} (1-G_{i,j})x_{t,k,i,j} \le \eta_{\max}, \quad t\in\mathcal{T}^{+}. \tag{17}$$

(17) represents the total number of pieces exchanged between groups at time t should be not greater than $\eta_{\max}$. In addition, we also need to ensure that the first objective, i.e., minimization of the average file download time among peers, should be satisfied, and thus the following constraint is required:

$$\tau = \tau^*,$$

where τ^* is the minimum average file download time, which can be obtained by solving the P_1.

As in P_1, P_2 is also ILP.

TABLE II: Evaluation models.

Communication Model	Intra group	Inter group
Relaxed TFT model	w/o TFT	with TFT
Strict TFT model	with TFT	with TFT
No TFT model	w/o TFT	w/o TFT

IV. NUMERICAL EVALUATION

A. Evaluation scenario

In this paper, we focus on the analysis of the optimal piece flow under the TFT-based P2P file distribution with the proximity-based TFT relaxation. Since the optimization problems P_1 and P_2 are ILP as mentioned in Section III, we can directly solve them the existing solver CPLEX when the system scale is relatively small. Considering the evaluation models used in the previous work [5]–[7], we prepare a default evaluation scenario as follows. Suppose that one server distributes eight pieces to eight peers, i.e., $(N_{\mathrm{D}} = 1, N_{\mathrm{P}} = 8, M = 8)$. In general, the inter-group upload capacity of the server 0, C_0^{out}, tends to be the bottleneck with increase of the number of peers, i.e., $C_0^{\mathrm{out}} \ll N_{\mathrm{P}}$, and thus C_0^{out} is set to be 2. To focus on the fundamental characteristics of the proximity-based TFT relaxation, we consider two groups $(G = 2)$with the same number of homogeneous peers $i \in \mathcal{N}_{\mathrm{P}}$ whose inter-group upload capacity is identical, i.e., $C_i^{\mathrm{out}} = C_0^{\mathrm{out}} = 2$. The intra-group upload capacity of peer $i \in \mathcal{N}_{\mathrm{P}}$, C_i^{in}, is set to be large enough not to be the bottleneck of the file distribution.

In addition to the two-stage ILP model, which we call *relaxed TFT model*, we also evaluate two models: *strict TFT model* and *no TFT model*, for comparison purpose. Table II summarizes the relationship between the three kinds of evaluation models and the TFT constraint. The strict TFT model is the same as the model in [5], where all leechers are strictly selfish and the TFT constraint is required between any pair of leechers, regardless of the communication type (i.e., intra-group communication and inter-group communication). In the no TFT model, all leechers are assumed to be altruistic, and thus the TFT constraint is not adopted to any pair of leechers, regardless of the communication type. The no TFT model and the strict TFT model will show the upper bound and lower bound of the system performance, respectively. In addition, these two models can also be regarded as two extreme cases of the relaxed TFT model because the no TFT model (resp. strict TFT model) is equivalent to the relaxed TFT model with $G = 1$ (resp. $G = N_{\mathrm{P}}$).

In what follows, we evaluate the system performance in terms of two criteria. First one is the average file download time among peers, $\overline{\tau}^*$, which is obtained by solving P_1.

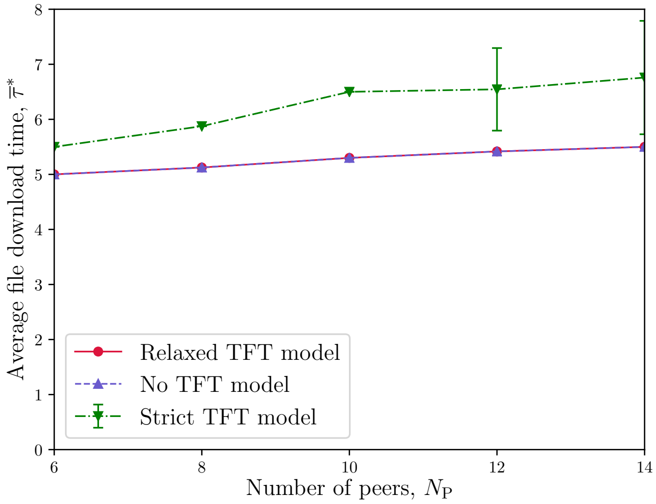

Fig. 1: Impact of the number of peers on average file download time ($M = 8, G = 2$).

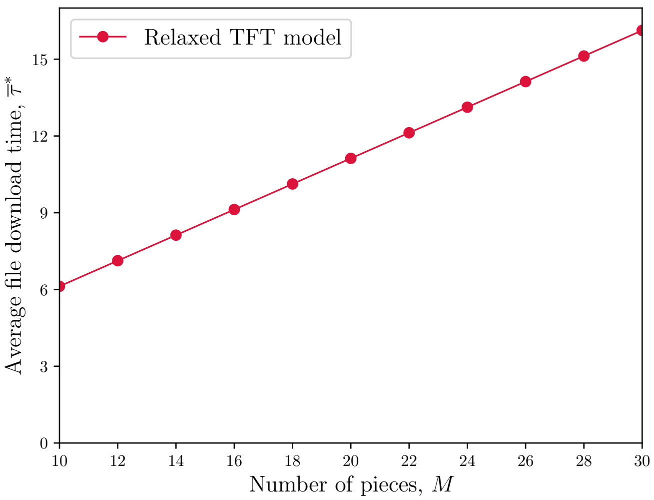

Fig. 2: Impact of the number of pieces on average file download time ($N_{\mathrm{P}} = 8, G = 2$).

Second one is the peak traffic $\eta^*_{\max}$, which is obtained by solving P_2. The following results are basically the optimal solutions obtained by CPLEX. In some cases, we only obtain the sub-optimal solutions, due to the computation complexity, and show the range of the lower bound and upper bound, which can also be obtained by CPLEX.

B. Impact of system scale on average file download time

In this section, we focus on how the system scale, i.e., the number of peers, N_{P}, and that of pieces, M, affects the average file download time in each model. Fig. 1 shows the transition of the average file download time $\overline{\tau}^*$ with the increase of the number of peers N_{P} when $M = 8$ and $G = 2$. We first observe that the relaxed TFT model show the same performance as the no TFT model, and thus it achieves the lower bound. This desirable phenomena come from the following optimal piece flow. The server 0 with $C^{\mathrm{out}}_0 = 2$ first distributes two different pieces b_1 and b_2 to the two peers k and l, each of which belongs to a different group, at $t = 1$. At the next time step $t = 2$, peers k and l speedily transfer b_1 and b_2 to their group members with sufficiently large intra-group upload capacity, i.e., $C^{\mathrm{in}}_k = C^{\mathrm{in}}_l = 10$. At the same time, they can also exchange b_1 and b_2 with each other through inter-group communication with the upload capacity $C^{\mathrm{out}}_k = C^{\mathrm{out}}_l = 2$. Note that the server 0 can also send the next two pieces to the two groups at $t = 2$. As a result, at $t = 3$, two groups can effectively exchange the pieces b_1 and b_2 because of the abundant pairs of peers satisfying the TFT constraint. The optimal piece flow emerges from the repetition of this effective cycle.

On the other hand, the average file download time of the strict TFT model, which is the upper bound, faster grows with the increase of the number of peers, compared with other two models. This is because peer cannot have enough pieces to exchange with others, due to the bottleneck of the inter-group upload capacity of the server 0.

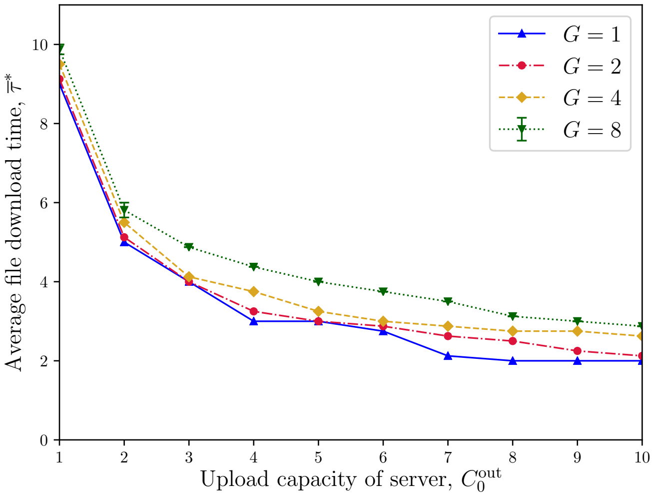

Fig. 3: Impact of the number of groups on average file download time ($N_{\mathrm{P}} = 8, M = 8$).

Next, we focus on the impact of the number of pieces M on the average file download time. Fig. 2 illustrates the relationship between the number of pieces, M, and the average file download time $\overline{\tau}^*$, in case of the relaxed TFT model with $N_{\mathrm{P}} = 8$ and $G = 2$. We observe the average file download time linearly increases with M, i.e., the content size. This indicates that the above-mentioned effective cycle can be achieved in the relaxed TFT model.

C. Impact of number of groups on average file download time

As mentioned in Section IV-A, the no TFT model and the strict TFT model are special cases of the relaxed TFT model, depending on the number of groups, G. In other words, the system performance of the relaxed TFT model will change according to G.

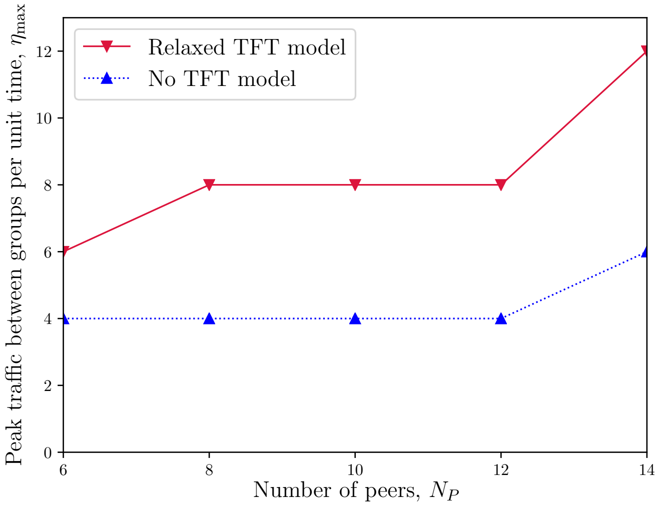

Fig. 4: Impact of the number of peers on peak traffic between groups ($M = 8, G = 2$).

Fig. 3 depicts the relationship between the number of groups, G, and the average file download time $\overline{\tau}^*$ when $N_{\mathrm{P}} = 8, M = 8$, and the inter-group upload capacity of the server, C_0^{out}, increases. We set to be $G = \{1, 2, 4, 8\}$ such that each group size is identical. For instance, the number of peers in each group becomes two when $G = 4$. As mentioned in Section IV-A, the relaxed TFT model with $G = N_{\mathrm{P}} = 8$ (resp. $G = 1$) is equivalent to the strict (resp. no) TFT model. As we expected, the average file download time of the relaxed TFT model increases with the increase of the number of groups, G. However, we also observe that the results of $G = 2$ and $G = 4$ can be competitive with those of $G = 1$, that is the lower bound, if the inter-group upload capacity of the server 0, C_0^{out}, is relatively small, e.g., $C_0^{\mathrm{out}} = [1, 3]$. In actual systems, the inter-group upload capacity of the server tends to be the bottleneck, and thus we can expect that the TFT constraint in the inter-group communication does not so much degrade the average file download time, with the help of the speedy intra-group communication. The TFT constraint in the inter-group communication can contribute to establishing the relationship of mutual trust between anonymous peers in different groups.

D. *Impact of system scale on peak traffic between groups*

Finally, we analyze the impact of system scale on the peak traffic between groups, i.e., $\eta_{\max}$. Fig. 4 shows the relationship between the number of peers, N_{P}, and the peak traffic between groups, $\eta_{\max}$, in case of $M = 8$ and $G = 2$. In Fig. 4, we only compare the relaxed TFT model with the no TFT model by considering the fact that they achieves the same performance in terms of the average file download time, as shown in Section IV-B. We observe that the results of the relaxed TFT model are twice as those of the no TFT model, except for $N_{\mathrm{P}} = 6$. This is because the relaxed TFT model requires the equivalent piece exchange in the inter-group communication, which is not necessarily required in the no TFT model.

V. CONCLUSION

In this paper, we have first modeled the TFT-based P2P file distribution with the proximity-based TFT relaxation where inter-group communications are only limited by the TFT constraint. We have formulated the optimal piece flow determination problem as the two-stage ILP that minimizes the average file download time among peers and the peak traffic among groups in this order. Through the numerical results, we have showed that the relaxed TFT model can reduce the average file download time up to about 18.3% compared with the strict TFT model in case of $M = 8, G = 2$. In addition, the relaxed TFT model can be competitive with the no TFT model in terms of the average file download time when the upload capacity of the server is bottleneck. Finally, we has also demonstrated that the relaxed TFT model requires about twice peak traffic between groups compared with the no TFT model.

ACKNOWLEDGMENTS

This work was supported in part by KDDI Foundation Research Grant Program, JSPS KAKENHI (A) under Grant 19H01103, and JSPS KAKENHI (C) under Grant 19K11942.

REFERENCES

[1] C. Carbunaru, Y. M. Teo, B. Leong, and T. Ho, "Modeling Flash Crowd Performance in Peer-to-Peer File Distribution," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2617–2626, Oct. 2014.
[2] "Deploy and Update Windows10," available at https://docs.microsoft.com/en-us/windows/deployment/.
[3] F. Azzedin and M. Yahaya, "Modeling BitTorrent Choking Algorithm Using Game Theory," *Future Generation Computer Systems*, vol. 55, no. 2, pp. 255–265, 2016.
[4] "BitTorrent," available at https://www.bittorrent.com.
[5] M. Hasegawa, M. Sasabe, and T. Takine, "Analysis of Optimal Scheduling in Tit-for-Tat-Based P2P File Distribution," *IEICE Transactions on Communications*, vol. 97, no. 12, pp. 2650–2657, 2014.
[6] M. Sasabe, "Analysis of Optimal Piece Flow in Tit-for-Tat-Based P2P Streaming," *Computer Networks*, vol. 139, no. 7, pp. 60–69, 2018.
[7] ——, "Topological Influence on Optimality of Tit-for-Tat based P2P Content Distribution," *Peer-to-Peer Networking and Applications*, pp. 1–12, May 2019.
[8] "IBM ILOG CPLEX Optimizer," available at https://www.ibm.com/analytics/cplex-optimizer.
[9] D. K. Goldenberg, L. Qiuy, H. Xie, Y. R. Yang, and Y. Zhang, "Optimizing Cost and Performance for Multihoming," in *Proc. of the ACM SIGCOMM Computer Communication Review*, 2004, pp. 79–92.
[10] E. de Souza e Silva, R. M. Leão, D. S. Menasché, and D. Towsley, "On the Scalability of P2P Swarming Systems," *Computer Networks*, vol. 151, no. 3, pp. 93–113, 2019.
[11] R. L. Xia and J. K. Muppala, "A Survey of BitTorrent Performance," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 140–158, February 2010.
[12] S. L. Blond, A. Legout, and W. Dabbous, "Pushing BitTorrent Locality to the Limit," *Computer Networks*, vol. 55, no. 3, pp. 541–557, 2011.
[13] N. Magharei, R. Rejaie, I. Rimac, V. Hilt, and M. Hofmann, "ISP-Friendly Live P2P Streaming," *IEEE/ACM Transactions on Networking*, vol. 22, no. 3, pp. 244–256, Feb. 2014.
[14] X. Zhang, N. Wang, Y. Cao, L. Peng, and H. Meng, "A Stochastic Analytical Modeling Framework on ISP–P2P Collaborations in Multidomain Environments," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2320–2331, Sep. 2018.
[15] M. A. D'Alessandro Costa and M. Gonçalves Rubinstein, "Performance Analysis of a Locality-Aware BitTorrent Protocol in Enterprise Networks," *Peer-to-Peer Networking and Applications*, pp. 1–12, Aug. 2018.
[16] D. Chen, R. G. Batson, and Y. Dang, *Applied Integer Programming*. Wiley, 2010.