Balancing Performance and Fairness in P2P Live Video Systems

Di Wu†, Yi Liang†, Jian He‡, Xiaojun He†
†Dept. of Computer Science, Sun Yat-Sen University, Guangzhou, China
‡Dept. of Electronics & Information Engineering, Huazhong University of Science & Technology, China
Email: {wudi27@mail, liangyi9@student, hejian9@mail2}.sysu.edu.cn, heixj@ hust.edu.cn

Abstract—Measurement studies of popular P2P live video systems reveal that there exists extreme unfairness among peers in the swarm. Such kind of unfairness will provide disincentives to altruistic super peers and encourage free riding behaviors in the system. It is essential for video service providers to take fairness into consideration when designing their systems. In this paper, we develop a simple model of P2P live video systems to understand the fairness problem from a theoretic perspective. We identify the fundamental tradeoff between fairness and performance, and propose a semi-distributed algorithm based on subgradient method to tune the P2P live video system towards the optimal fairness while still maintaining the targeted universal streaming rate. We also conduct extensive trace-driven simulations to validate the effectiveness of our proposed algorithm. The simulation results show that our algorithm can guide the system towards the optimal fairness quickly without degrading the streaming performance at the same time.

I. INTRODUCTION

LIVE video streaming has become one of the most popular applications over the Internet and accounts for a significant fraction of Internet traffic [1]. To avoid the prohibitive server bandwidth cost of the client/server model, peer-to-peer (P2P) paradigm has been adopted as an effective solution to deliver videos with significantly lower infrastructure and bandwidth cost. In the past decade, we have witnessed several successful large-scale deployments of P2P live video systems, such as PPLive [2], PPStream [3], UUSee [4], Sopcast [5], Coolstreaming [6], etc.

To improve the overall system performance and better user experience, most of existing systems exploit upload bandwidth of super peers (e.g., campus peers with Ethernet connections) extensively to compensate the low upload capacity of weak peers (e.g., residential peers with ADSL/modem connections); however, fairness among peers has generally been ignored by system designers. Based on measurement results in [7], [8], there exists extreme unfairness among peers in popular P2P live video systems. In those systems, in spite that both super peers and weak peers receive video streams of the same rate, the upload rate of a super peer can be almost ten times of that of a weak peer. Such kind of unfairness will provide disincentives to super peers and encourage free-riding behaviors in the system. Most commercial P2P live video applications adopt a proprietary closed-source design, hence it is difficult to reverse-engineer the system and modify the protocols. Free riders in the P2P live video systems generally resort to third-party rate-limiting softwares (e.g., personal firewalls) for bandwidth limitation.

To address the free-riding problem, there has been plenty of research work [9], [10], [11], [12], [13] on providing incentives in P2P video streaming systems. The purpose of most incentive designs is to increase upload bandwidth contribution from peers and thus improve the overall system performance. However, the problem of how to fairly exploit peer resources have not received enough attention. There also lacks quantitative analysis of the fairness level in a P2P video streaming system, and it is unclear how to balance performance and fairness during system design.

In this paper, we develop a theoretic model to formally analyze the fairness problem in the P2P live video system, and identify the intrinsic tradeoff between performance and fairness. To the best of our knowledge, our paper is the first to model and analyze such kind of tradeoff in P2P live video systems. We also propose a practical semi-distributed algorithm to achieve the optimal fairness under a certain streaming rate constraint. Our main contribution in this paper can be summarized as below:

- We develop a simple theoretic model of P2P live video systems that takes both node heterogeneity and peer churn into account. With the above model, we formally analyze the fairness level that can be achieved in a P2P live video system and further derive the optimal values of central metrics, such as fairness index, streaming rate, etc.
- We identify the tradeoff between fairness and performance, and design a semi-distributed algorithm based on subgradient method [14] to achieve the optimal fairness while still maintaining the targeted streaming rate. The algorithm is practical and can be easily integrated into the existing P2P video streaming systems.
- We conduct extensive trace-driven simulations to validate the effectiveness of our proposed algorithm. Our simulation results show that our algorithm can well balance fairness and performance in a P2P live video system. The whole system can rapidly converge to the max-min fairness without degrading the streaming quality.

Different from previous work on incentive designs for P2P streaming systems, our purpose in this paper is not to encourage all peers to upload as much as they can. Instead, we aim at designing fair bandwidth allocation algorithms that can be implemented on individual peers to enhance the fairness.
level in the system. The work on incentive mechanisms is complementary to our algorithm and can be integrated with our work together. A proper incentive mechanism can encourage peers to report their information honestly and upload as much as they can. Our algorithm can tell a peer the targeted upload rate it should maintain to guarantee the system fairness.

The remainder of this paper is organized as follows: In Section II, we review prior work on the design of incentive mechanisms and theory papers related to P2P live video systems. In Section III, we develop a simple model to analyze fairness in a P2P live video system and derive the optimal fairness under a given streaming rate. In Section IV, we propose a distributed algorithm to achieve the max-min fairness in a P2P live video system. In Section V, we conduct extensive simulations to evaluate our proposed algorithm. Finally, we summarize our work in Section VI.

II. RELATED WORK

P2P live video streaming has attracted lots of research activities in the past years. Most of previous work focused on the design, measurement and improvement of P2P streaming systems from the performance perspective. The survey papers [15], [16], [17] summarized the current research and development progress in the fields.

A closely related area is the design of incentive mechanisms. Fairness is a system-wide property to evaluate how fair the contribution of each peer is. Incentive mechanisms are mostly used to defend against free-riding and encouraging peer contribution. The use of incentives can improve the system performance but possibly decrease the fairness in the system, for example, when exploiting the bandwidth of super peers heavily. Researchers have conducted quite a few studies on the design of incentives for P2P streaming systems. Liu et al. [18] proposed an incentive mechanism based on bilateral data exchange for layered P2P video systems, in which peers exchange their resources only when both of them have the resources interested by their counterparts. Lin et al. [13] considered the design of tit-for-tat style incentive mechanisms for MDC-based P2P streaming systems. Tan et al. [19] introduced the concept of virtual currency into P2P streaming systems, which can be used among peers and increase the chance of resource distribution. Shi et al. [20] adopted a reputation-based design to determine service levels among peers according to their interaction history, and avoided the use of currency managers in the micropayment-based system. Li et al. [9] proposed a taxation-based mechanism for P2P streaming systems with layered topology and incentivized high-bandwidth peers to contribute more. Hu et al. [21] proposed a taxation mechanism to achieve the balance between social welfare and individual utility in layered P2P streaming systems. Wang et al. [11] proposed to use advertisements as an incentive to encourage peer uploading in operational P2P streaming systems. Chuan et al. [12] designed a discriminative second price auction strategy for P2P VoD streaming systems, which let budget-constrained peers bid for desired blocks from their neighbors so that optimal media block scheduling can be achieved. In spite that different mechanisms have been proposed to date, however, there is no guarantee on how fair the system can achieve under a given incentive mechanism. This brings forth the necessity of the theoretical analysis of fairness problem in the P2P live video systems to reveal the fundamental trade-offs between performance and fairness.

Theoretical treatments of P2P live video systems have been in progress well. Kumar et al. [22] provided a stochastic fluid model to expose the fundamental characteristics and limitations of single-channel P2P streaming systems. Massoulie et al. [23] studied the problem of efficient decentralized broadcasting in both edge-capacitated and node-capacitated networks. Liu et al. [24] derived the performance bounds in terms of minimum server load, maximum streaming rate, and minimum tree depth under different peer selection constraints. Wu et al. [25] developed a queueing network model to analyze the performance of multi-channel P2P streaming systems. In [26], Zhou et al. analyzed the impact of different chunk-scheduling strategies on the streaming quality. In [27], Liu studied the minimum delay that can be achieved in the P2P streaming systems. However, few work analyzed the fairness problem in P2P streaming systems, and it is still unclear for designers to understand how to optimize the fairness among peers for video streaming services. Only Fan et al. [28] analyzed the impacts of optimistic/regular unchoking strategies on the fairness level in the BitTorrent systems, but their work focused on the BitTorrent file-sharing systems, instead of P2P live streaming systems.

Our work differs from the previous work in two-folds: 1) instead of solely analyzing performance issues, we focus on analyzing the fairness problem in P2P live video systems, and identifying the underlying tradeoffs between performance and fairness in the design space; 2) We also proposed a distributed algorithm to tune the system towards the optimal fairness, and our algorithm is theoretically founded and able to guarantee the streaming performance simultaneously.

III. FAIRNESS ANALYSIS OF P2P LIVE VIDEO SYSTEMS

In this section, we develop a theoretic model to analyze the fairness problem in P2P live video systems. Our model takes peer churn and multi-class peers with different bandwidth capacities into account. With this model, we can expose the fundamental tradeoff between fairness and performance in the P2P live video systems.

To simplify our model, the following reasonable assumptions are made in this paper: (1) considering the prevalence of single-layer video streaming, only single-layer video is considered in our model. A peer can watch a video channel if and only if it can receive video stream at the full rate. (2) peers are categorized into multiple classes according to their upload bandwidth and peers in the same class are assumed to have the same upload rate. (3) the bottleneck is assumed not at the core of the network, but at the network edge. It means that the streaming performance is only constrained by upload capacity of peers and servers. (4) peer arrival is assumed to follow a Poisson process, but peer sojourn time can follow an arbitrary distribution.

In our model, we consider a dynamic P2P live video system with continuous peer join and departure (as shown in Figure 1).
The arrival rate of peers into the whole system is denoted by \( \lambda \). The arriving peers are classified into \( K \) types according to their upload capacity. Let \( U_k \) denote the upload capacity of the \( k \)-type peers \((k = 1, \ldots, K)\), which satisfies \( U_1 \leq U_2 \leq \ldots \leq U_K \). Assume that with probability \( p_k \) that a new arriving peer is a \( k \)-type peer \((\sum_{k=1}^K p_k = 1)\). Thus, the arrival rate of the \( k \)-type peers is given by \( \lambda_k = p_k \lambda \).

To evaluate the performance of a P2P live video system, the maximum achievable streaming rate (denoted by \( R \)) is adopted as the performance metric, which is defined as the maximum streaming rate that the system can sustain and universal streaming [22] is achieved among all peers.

The Jain’s fairness index [29] is used as the metric to evaluate the fairness level in a P2P live video system. Our definition of fairness takes both upload rate and download rate into account. The definition of Jain’s fairness index \( F \) is given as follows:

\[
F = \frac{\left( \sum_{i=1}^N x_i/y_i \right)^2}{N \sum_{i=1}^N (x_i/y_i)^2}
\]

where \( x_i, y_i \) are the upload rate and download rate of the \( i \)-th peer and \( N \) is the total number of peers in the P2P live video system. The above definition can be understood as the fairness index of sharing ratios of peers. For single-layer video streaming, as all peers have the same download rate (i.e., video streaming rate) when universal streaming is achieved, the variables of download rate \( y_i \) can be eliminated from the definition of fairness. In this case, the definition of fairness can be simplified as below:

\[
F = \frac{\left( \sum_{i=1}^N x_i \right)^2}{N \sum_{i=1}^N x_i^2}
\]

Based on our assumption that all the peers of the same type have the same upload rate, the fairness index can be further transformed into

\[
F = \frac{\left( \sum_{k=1}^K p_k u_k \right)^2}{\sum_{k=1}^K p_k u_k^2},
\]

where \( u_k \) is the upload rate of \( k \)-type peers, and \( K \) is the number of peer types in the system. All the notations used in this paper are summarized in Table I. Next, we will study a P2P live video system based on the above model.

### Table I: Notation

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( u_s )</td>
<td>the server upload rate</td>
</tr>
<tr>
<td>( U_k )</td>
<td>the maximum upload capacity of ( k )-type peers;</td>
</tr>
<tr>
<td>( u_k )</td>
<td>the upload rate of ( k )-type peers, ( 0 \leq u_k \leq U_k );</td>
</tr>
<tr>
<td>( x_i )</td>
<td>the upload rate of the ( i )-th peer;</td>
</tr>
<tr>
<td>( y_i )</td>
<td>the download rate of the ( i )-th peer;</td>
</tr>
<tr>
<td>( K )</td>
<td>the number of peer types in the system</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>the total arrival rate of peers;</td>
</tr>
<tr>
<td>( p_k )</td>
<td>the probability that a new arriving peer is a ( k )-type peer;</td>
</tr>
<tr>
<td>( \lambda_k )</td>
<td>the arrival rate of the ( k )-type peers, ( \lambda_k = p_k \lambda );</td>
</tr>
<tr>
<td>( N )</td>
<td>the total number of peers in the system under steady-state;</td>
</tr>
<tr>
<td>( N_k )</td>
<td>the number of ( k )-type peers in the system;</td>
</tr>
<tr>
<td>( T )</td>
<td>the average sojourn time in the system;</td>
</tr>
<tr>
<td>( R )</td>
<td>the maximum achievable streaming rate.</td>
</tr>
<tr>
<td>( F )</td>
<td>the fairness index in the P2P live video system.</td>
</tr>
</tbody>
</table>

### A. Optimal Performance

In this section, we first try to answer the following important questions: what is the optimal performance in a P2P live video system? how should a system approach the state of optimal performance? what is the fairness level when the system achieves optimal performance? For simplicity, we consider a case when the server upload bandwidth is much higher than that of any peer, i.e., \( u_s > U_K \), where \( u_s \) is the server upload rate. It is not difficult to obtain the following theorem:

**Theorem III.1** The maximum achievable streaming rate in a P2P live video system is given by

\[
R_{\text{perf}} = \frac{u_s}{T} + \sum_{k=1}^K p_k U_k
\]

and the fairness index when achieving the maximum performance is given by

\[
F_{\text{perf}} = \frac{\left( \sum_{k=1}^K p_k U_k \right)^2}{\sum_{k=1}^K p_k U_k^2}.
\]

To achieve the optimal performance, every peer should upload at its maximum upload capacity.

Theorem III.1 implies that only when every peer uploads at its full capacity can the optimal performance be achieved.

### B. Optimal Fairness

Next, we proceed to study what is the optimal fairness that can be achieved in a P2P live video system and how to approach the optimal fairness state. Starting from the fairness metric, we can obtain the following theorem:

**Theorem III.2** The optimal fairness index that can be achieved is

\[
F_{\text{fair}} = 1
\]

and the maximum achievable streaming rate when optimal fairness is achieved is

\[
R_{\text{fair}} = \frac{u_s}{T} + U_1.
\]
To achieve the optimal fairness in the P2P live video system, all the peers should have the same upload rate.

Theorem III.2 points out that the optimal fairness is achieved only when all peers upload at the same rate. However, the system performance becomes pretty bad when optimal fairness is achieved, as the maximum upload rate of peers is constrained by upload capacity of the weakest peer in the P2P video system.

C. Optimizing Fairness under a Certain Streaming Rate

The above sections describe two extreme cases in which either performance or fairness is optimized. In this section, we consider a more general and realistic case and answer how to maximize the fairness while still maintaining a targeted universal streaming rate.

In order to optimize the fairness index under a given universal streaming rate $r$, $R_{fair} < r < R_{perf}$, we formulate the optimization problem as follows:

Maximize $F = \frac{\sum_{k=1}^{K} p_k u_k}{\left(\sum_{k=1}^{K} p_k u_k^2\right)^{\frac{3}{2}}}$
subject to

$0 \leq u_k \leq U_k$
$\sum_{k=1}^{K} p_k u_k + \frac{u_s}{T} = r$
$1 \leq k \leq K$

where the objective function is the fairness index to be optimized, the first constraint indicates the maximum upload capacity that a peer can allocate, and the second constraint is the necessary condition to achieve the universal streaming rate $r$ in the system under steady state.

The above problem can be transformed into a nonlinear optimization problem as below:

Minimize $f(u) = \sum_{k=1}^{K} p_k u_k^2$

s.t.

$-u_k \leq 0$
$u_k - U_k \leq 0$
$\sum_{k=1}^{K} p_k u_k - r' = 0$
$1 \leq k \leq K$

where $u = (u_1, \ldots, u_K) > 0$ and $r' = r - \frac{u_s}{T}$. The Lagrangian multiplier of the above nonlinear optimization problem can be given as follows:

$L(u, \alpha, \beta, \nu) = f(u) + \sum_{k=1}^{K} \alpha_k g_1^{(k)}(u_k) + \sum_{k=1}^{K} \beta_k g_2^{(k)}(u_k) + \nu h(u)$

where $g_1^{(k)}(u_k) = -u_k$, $g_2^{(k)}(u_k) = u_k - U_k$ and $h(u) = \sum_{k=1}^{K} p_k u_k - r'$. By Karush-Kuhn-Tucker(KKT) conditions[14], we can solve the above nonlinear optimization problem and obtain the following theorem:

**Theorem III.3** The optimal fairness achieved under a given streaming rate $r$ is given by

$$F = \frac{(r - \frac{u_s}{T})^2}{\sum_{k=1}^{m} p_k U_k^2 + \sum_{k=m+1}^{K} p_k u_k^2}$$

where $m$ is a threshold less than $K$ and $u$ satisfies $U_m \leq u \leq U_{m+1}$. To optimize the fairness under a given streaming rate $r$, all the peers whose type is no greater than $m$ should upload at their full capacity, while all the peers whose type is greater than $m$ should upload at the same rate $u$.

**Corollary III.4** The bandwidth allocation when optimal fairness is achieved under a given streaming rate corresponds to max-min fairness [30].

Under max-min fairness, upload bandwidth of peers with smaller upload capacity should be exploited as much as possible, if they wish to receive the same streaming rate as that of other super peers.

D. Tradeoff between Fairness and Performance

In the following, we will investigate the relationship between fairness and performance based on the above mathematical analysis. To better our description, we first use a P2P live video system that contains five types of peers as an example to show the conflict between performance and fairness. Peers in different types have different upload capacities and their bandwidth distribution is given in Table II. The server upload bandwidth is set as 1Mbps.

<table>
<thead>
<tr>
<th>Peer Type</th>
<th>1-type</th>
<th>2-type</th>
<th>3-type</th>
<th>4-type</th>
<th>5-type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upload Capacity(Kbps)</td>
<td>128</td>
<td>384</td>
<td>512</td>
<td>768</td>
<td>1024</td>
</tr>
<tr>
<td>Percentage(%)</td>
<td>15.0%</td>
<td>25.0%</td>
<td>35.0%</td>
<td>15.0%</td>
<td>10.0%</td>
</tr>
</tbody>
</table>

**TABLE II:** Bandwidth distribution of peers

![Fig. 2: Tradeoff between fairness and performance](image)

In our numerical study, we calculate the values of optimal fairness under different streaming rates according to theorems derived in the previous section and plot the obtained results in Fig. 2. From the figure, we can observe that there exists
a clear tradeoff between fairness and performance in the P2P live video system. When the fairness metric approaches the optimality (that is, fairness = 1), all peers in the system choose to upload at the same rate and thus degrade the performance. The universal streaming rate corresponds to the upload rate of the weakest peers (i.e., streaming rate = 128 Kbps). On the contrary, when the optimal performance is achieved (streaming rate = 512 Kbps), the fairness is not good as well (fairness = 0.81). For a certain streaming rate \( r \) between 128 Kbps and 512 Kbps, there is a design space that allows us to tune the fairness level of a P2P live video system.

![Fig. 3: Design Space of a P2P live video System](image)

We further consider a generalized case and illustrate the range between the worst fairness and the best fairness under a given rate streaming \( r \) in Fig. 3. The design space for engineers to tune the fairness level in the system is widened when the streaming rate is lower. For a given streaming rate \( r \) between \( R_{perf} \) and \( R_{fair} \), on one extreme we can optimize the system to achieve max-min fairness, in which low-bandwidth peers upload at their full capacity, while high-bandwidth peers upload at the same rate; on another extreme, we can achieve the worst fairness in the system, in which only a few super peers take over all the distribution responsibility and weak peers make no contribution.

To be fair for different types of peers, when bandwidth resource is abundant to support universal streaming in the system, it is not necessary to further exploit the bandwidth of super peers. Instead, it is more desirable to perform fair bandwidth allocation among peers. For designers of future P2P streaming systems, it is essential to achieve a higher level of fairness among peers in addition to improving performance.

IV. TUNING P2P LIVE VIDEO SYSTEMS TOWARDS OPTIMAL FAIRNESS

In the previous section, we perform analysis of the fairness problem in the P2P streaming system and derive the corresponding peer bandwidth allocation strategies under each scenario. In this section, we will propose a practical algorithm to implement fair bandwidth allocation on individual peers and tune the system towards optimal fairness while still maintaining a certain universal streaming rate. The problem on free-riding and cheating can be solved by introducing proper incentive mechanisms as stated in Sec II, which encourage peers to report their information honestly and upload as they can. But the design of a proper incentive mechanism is beyond the scope of this paper. The main purpose of our paper is to direct each peer to upload at a reasonable level to guarantee the system fairness.

Theorem III.3 indicates that, to achieve the max-min fairness under a certain streaming rate, we need to let peers with bandwidth capacity lower than a threshold upload at full capacity and peers with bandwidth capacity higher than the threshold upload at the same rate. Such max-min fairness can be achieved by using the progressive water-filling algorithm [31]. When executing the water-filling algorithm, the upload rates of all peers are initially set as zero and then grow at the same step, until one or more peers’ upload capacity limits are hit. The upload rates of peers whose upload capacity is fully utilized will not increase any more, but the rates of other peers continue to increase. The increase stops when the amount of allocated bandwidth is enough to achieve the targeted streaming rate. Such kind of bandwidth allocation is max-min fair.

An example to illustrate how the water-filling algorithm works is shown in Fig. 4. In the example, suppose that there are three peers with different upload capacities in the system. Each peer is represented by a bin. The water in each bin indicates the allocated upload bandwidth and the dashed line indicates the maximum upload capacity. As shown in Fig. 4(a), initially all the bins are empty and water is filled into three bins at the same rate. When the bin \( W1 \) is full (see Fig. 4(b)), we stop to fill water into \( W1 \) but continue to fill the bins \( W2 \) and \( W3 \) (see Figure 4(c)). In Fig. 4(d), the water filling process is stopped when the amount of allocated bandwidth is enough to satisfy the targeted streaming rate. It is not difficult to implement the above water-filling algorithm by centralized algorithms but it suffers from the scalability problem. A distributed solution is more preferable in the environment of P2P systems.

In the next section, we will propose a semi-distributed algorithm to optimize fairness in a P2P live video system. Using the Lagrangian relaxation method, the fairness optimization problem can be solved by a semi-distributed algorithm implemented on individual peers.

A. Semi-distributed Algorithm to Optimize Fairness

To optimize the fairness metric \( F \) while maintaining a streaming rate \( r \), we need to solve the optimization problem below.

Minimize  
subject to  
\[
\begin{align*}
    f(x) &= \sum_{i=1}^{N} x_i^2 \\
    0 &\leq x_i \leq U_i, i = 1, \cdots, N \\
    \frac{\sum_{i=1}^{N} x_i + u_s}{N} &= r 
\end{align*}
\]  
(3)
Let \( r' = r - \frac{r}{N} \). The Lagrangian of the above problem can be derived as below:

\[
L(x, \alpha, \beta, \nu) = \sum_{i=1}^{N} x_i^2 - \sum_{i=1}^{N} \alpha_i \cdot (x_i) + \sum_{i=1}^{N} \beta_i \cdot (x_i - U_i) + \nu\left(\sum_{i=1}^{N} x_i - r'\right)
\]

\[
= \sum_{i=1}^{N} (x_i^2 - \alpha_i x_i + \beta_i x_i + \nu x_i)
\]

where \( L_i(x_i, \alpha_i, \beta_i, \nu) = x_i^2 - \alpha_i x_i + \beta_i x_i + \nu \frac{x_i}{N} \) is the \( i \)-th Lagrangian to be minimized by the \( i \)-th peer. Given \( \alpha_i, \beta_i, \nu \), peer \( i \) aims to resolve the following optimizing problem:

\[
x_i^*(\alpha_i, \beta_i, \nu) = \arg\min x_i^2 - \alpha_i x_i + \beta_i x_i + \nu \frac{x_i}{N} \quad (4)
\]

By applying duality theory [14], the original problem can be transformed into the following dual problem:

Maximize

\[
g(\alpha, \beta, \nu) = \sum_{i=1}^{N} \left( g_i(\alpha_i, \beta_i, \nu) - \beta_i U_i \right) - \nu r'
\]

subject to \( \alpha_i \geq 0, \beta_i \geq 0 \) \quad (5)

where \( g_i(\alpha_i, \beta_i, \nu) = L_i(x_i^*, \alpha_i, \beta_i, \nu) \). The dual problem can be solved efficiently by using subgradient method as below:

\[
\alpha_i(t+1) = [\alpha_i(t) + \lambda \cdot (x_i)_+]_+ \quad (6)
\]

\[
\beta_i(t+1) = [\beta_i(t) + \gamma \cdot (x_i - U_i)_+]_+ \quad (7)
\]

\[
\nu(t+1) = \nu(t) + \theta \cdot \left( \sum_{i=1}^{N} x_i - r' \right) \quad (8)
\]

where \( t \) is the iteration index, \( \lambda, \gamma, \theta \) are sufficiently small positive step-sizes. The details of our proposed algorithm are provided in Algorithm 1.

### Algorithm 1 Semi-distributed fairness optimization algorithm

**Notation:**

- \( x_i \): the actual upload bandwidth of peer \( i \);
- \( \alpha_i, \beta_i \): the local Lagrange multipliers of peer \( i \);
- \( \nu \): the global Lagrange multiplier of peers;
- \( \lambda, \gamma, \theta \): the constant step sizes of each iteration;
- \( U_i \): the maximum upload bandwidth of peer \( i \);
- \( N \): the total number of peers in the system under steady-state;
- \( t \): the iteration index;

**Output:**

Upload bandwidth allocation of peers under optimal fairness state.

1. \( \alpha_i, \beta_i, \nu \) are initialized as zero;
2. \( \lambda, \gamma, \theta \) are initialized as a small positive value;
3. Each peer \( i \) locally resolves \( x_i^*(\alpha_i, \beta_i, \nu) \) according to (4) and sends the solution to the central coordinator;
4. Each peer \( i \) updates \( \alpha_i \) to be \([\alpha_i(t) + \lambda \cdot (x_i)_+]_+ \), and \( \beta_i \) to be \([\beta_i(t) + \gamma \cdot (x_i - U_i)_+]_+ \);
5. The coordinator updates \( \nu \) to be \( \nu(t) + \theta \cdot (\sum_{i=1}^{N} x_i - r') \) after receiving \( x_i^*(\alpha_i, \beta_i, \nu) \) from all peers according to (8), and broadcasts the updated value to all peers;
6. Set \( t \to t+1 \) and go to step 1 (until satisfying termination criterion).

### B. Practical Implementation of Semi-distributed Algorithm

Our algorithm can be easily integrated into existing P2P live video systems. For the practical implementation, a functional entity called coordinator is introduced for global price computation. The coordinator can be deployed either on a separate server or together with the tracker server. The semi-distributed algorithm is executed by message communication between peers and the coordinator. Three types of protocol messages are used in our implementation: (i) Bandwidth Message (BM), which contains the information about upload rate \( x_i \) of peer \( i \); (ii) Global Price Message (GPM), which indicates the global price \( \nu \) derived by the coordinator; (iii) Local Price Message (LPM), which contains the information about local prices \( \alpha_i \) and \( \beta_i \) of peer \( i \). In our protocol, a peer reports its recent upload rate by sending a BM message to the coordinator, and the coordinator broadcasts the global price \( \nu \) by sending a
GPM message to all peers. LPM messages are generated and processed locally. The messaging protocol of our algorithm can be illustrated by Fig. 5. Note that the term “price” actually refers to the Lagrange multiplier in our algorithm. The “local price” represents the Lagrange multiplier $\alpha_i$ and $\beta_i$ associated with individual peers locally, and the “global price” represents the Lagrange multiplier $\nu$, which is shared among all the peers in the system.

Fig. 5: Messaging model for the fairness optimization protocol

Algorithm 2 Semi-distributed fairness optimization algorithm executed on individual peers

Notation:
- $GMQ_i$: the global message queue of peer $i$;
- $LMQ_i$: the local message queue of peer $i$;
- $x_i$: the actual upload bandwidth of peer $i$;
- $\alpha_i, \beta_i$: the local prices of peer $i$;
- $GPM$: the global price message of the coordinator;
- $\nu$: the global price derived by the coordinator;

Output:
Peer $i$ updates local prices $\alpha_i$ and $\beta_i$ and sends the actual upload bandwidth $x_i$ to the coordinator.

1: Retrieve global price $\nu$ from the message $GPM$ stored in $GMQ_i$;
2: Retrieve local prices $\alpha_i$ and $\beta_i$ from $LMQ_i$;
3: $LMQ_i \leftarrow LMQ_i - \{\alpha_i, \beta_i\}$;
4: while the queue $GMQ_i$ is not empty do
5: Compute $x_i$ by solving $x_i^* (\alpha_i, \beta_i, \nu)$;
6: Update local prices $\alpha_i$ and $\beta_i$ according to (6) and (7) respectively;
7: Send the coordinator a BM message containing $x_i$;
8: $GPM_i \leftarrow \emptyset$;
9: $LMQ_i \leftarrow LMQ_i \cup \{\alpha_i, \beta_i\}$.
10: end while

Each peer $i$ maintains two queues for received messages: (1) $GMQ_i$, which stores messages received from the coordinator; (2) $LMQ_i$, which stores messages generated locally. The arrival of a new message will cause an event to be handled by the peer. If $GMQ_i$ is not empty, it means that the updated global price has not been processed by the peer. When peer $i$ detects $GPM$ messages left in the queue, it will compute the new upload rate $x_i$ and update the local prices $\alpha_i$ and $\beta_i$ accordingly. The updated value of $x_i$ is encapsulated in a $BM$ message and sent from peer $i$ to the coordinator. After each update, peer $i$ will clear the queue $GPM_i$ and replace the messages in the queue $LMQ_i$ with updated local prices $\alpha_i$ and $\beta_i$. For the coordinator, it only maintains one queue $BMQ$, which stores all the incoming $BM$ messages from peers. Note that the $BMQ$ queue only stores the latest $BM$ messages of each peer.

To cope with peer dynamics, time is divided into slots and the algorithm is executed periodically. At the beginning of each time slot, a peer sends an updated information message to the coordinator if there is any change of its upload rate. It also serves a heart-beat message to indicate the liveness of a peer, and therefore the coordinator can know the number of peers in the channel. In every time slot, after collecting the information of upload rates from all peers, the coordinator will update the global price $\nu$ based on our proposed algorithm. Then a $GPM$ message with the new price $\nu$ is broadcasted to all peers. At the end of each update, the coordinator will clear the queue $BMQ$. The details of our messaging protocol executed on peers and the coordinator are provided in Algorithm 2 and Algorithm 3.

With the increase of the number of peers in the system, the coordinator will become a potential bottleneck of the system, as it needs to broadcast the global price to every peer. We can exploit gossiping protocol [32] to relieve the load on the coordinator. Instead of sending the message via unicast to every peer, the coordinator only sends the GPM message to a small set of stable peers [33], who will later relay the GPM messages to other peers. Unlike the naive approach in Fig. 5, the coordinator adds a timestamp into the generated GPM message before delivering it. Peers interact with their neighbors to obtain the latest GPM message. They first compare the timestamps of their locally stored GPM messages and the GPM with a larger timestamp will be sent to the peer with a smaller timestamp. Upon receiving a new GPM message, the peer can update the global price accordingly. The gossiping protocol can greatly improve the efficiency of global price broadcasting. The coordinator can be relieved from the
load of message broadcasting.

The truthfulness of reported upload information can be guaranteed by various approaches, such as remote attestation [34], bandwidth estimation [35], etc. It is also feasible to utilize incentive mechanism to achieve honest information reporting. With a properly deployed incentive mechanism, peers can be encouraged to report their bandwidth honestly and upload as much as it can, but our algorithm can tell them how much they should upload in order to achieve the max-min fairness in the system. For example, when adopting layered taxation [9] as an incentive mechanism, peers who report high upload bandwidth can stay in a layer closer to the streaming source and have smaller playback latency. However, for each layer, there is a corresponding tax rate and every peer should pay such tax in order to receive streaming services. Peers in the upper layers must pay a higher tax than peers in the lower layers. Such a high tax can discourage peers from misreporting their upload bandwidth.

To integrate our algorithm with such an incentive mechanism, we can use our algorithm to determine the targeted upload rate in each layer and adjust the tax rate of the incentive mechanism for each layer accordingly. By collecting an appropriate amount of taxes from peers in different layers, it is possible to force each peer to upload at their targeted rate and achieve optimal fairness. The improvement of fairness is to avoid exploiting the bandwidth of super peers extensively, which play a key role for a P2P system. In reality, for a given video streaming rate \( r \), the system can target at maintaining a bit higher streaming rate \( r^* \) to increase resource index. With our mechanism, high-bandwidth peers will contribute more upload bandwidth, but the burden can be evenly distributed to all high-bandwidth peers.

If a peer really wants to contribute more bandwidth without degrading the system-wide fairness level, the problem can be solved by creating multiple “virtual nodes”. “Virtual node” is a common technique used in P2P systems for the purpose of load balancing, resource allocation, etc. In our proposed scheme, the use of virtual nodes is to ensure that system-wide optimal fairness can be achieved across all peers (including both real and virtual nodes). The maintenance overhead of virtual nodes is trivial compared with the huge volume of video traffic. For multiple virtual nodes hosted on one peer, they can share the global message queue and only need to maintain a local message queue individually. The control overhead can be further reduced by using compression techniques such as bloom filters. If not using virtual nodes, another possible solution is to tag peers according to their willingness to contribute. Peers who wish to make fair contribution will run our algorithm to set the targeted upload rate, while peers who are altruistic can simply set their maximum upload capacity as the targeted upload rate. However, in this case, system-wide optimal fairness cannot be achieved among all peers.

In reality, the execution of our algorithm may be impacted the correlation among peers when they share the same upload link. In this case, peers can change the reported upload bandwidth to their corresponding bandwidth share.
over time is shown in Fig. 6. It can be observed that high-bandwidth peers contribute the most in the initial period, but after a while the upload rates of high-bandwidth peers decrease significantly. On the contrary, the upload rates of middle-bandwidth and low-bandwidth peers continue to increase until reaching their upload capacity. Finally, the system enters the steady state around 300 seconds later, which also implies the convergence of our semi-distributed algorithm algorithm.

We further check the average utilization of upload bandwidth for different types of peers in Fig. 7. For a given peer, its utilization of upload bandwidth is defined as the percentage of its upload bandwidth utilized for P2P streaming. The water-filling algorithm is to calculate the targeted upload rate. It doesn’t mean upload rates of peers are really adjusted step-by-step from zero. In Fig. 7, we can observe that the high-bandwidth peers upload at their full capacity in the initial phase and then drop in afterwards. After an initial period of fluctuation, the utilization of middle-bandwidth and low-bandwidth peers converges to 100%. Instead, the utilization of high-bandwidth peers drops from 100% to 59%. The above observation indicates that our algorithm will not overload super peers and can improve the fairness level of the P2P streaming system. It also validates the statement in Theorem III.3. We find there exists a threshold $m = 2$ and all the peers whose type is no greater than two (i.e., low-bandwidth and middle-bandwidth peers in our experiment) should upload at their full capacity in order to maximize fairness in the system.

Fig. 8 shows the average streaming rates of peers in the system. It is found that our algorithm can guarantee that most peers receive the full streaming rate. The reason why a few peers cannot achieve the full streaming rate is caused by imperfect data scheduling. To let more peers receive the full video stream, we can allocate more bandwidth resources by increasing the targeted streaming rate a bit (e.g., $r+\epsilon$) and thus mitigate the stringent scheduling requirement due to resource deficiency[38].

Fig. 9 shows the evolution of fairness index in a P2P streaming system under three different streaming rates. It is observed that the fairness index becomes stable around 400 seconds for three different streaming rates. However, the achieved fairness indexes are not the same for different streaming rates. The higher the streaming rate is, the lower the fairness index is. It is due to the fundamental tradeoff between fairness and performance as identified in Sec III-D. As shown
in Fig. 2, the achieved optimal fairness (namely max-min fairness) decreases when the streaming rate increases.

To study the scalability of our algorithm, we plot the evolution of fairness index of P2P streaming systems at different scales in Fig. 10. In the experiments, we have tested a system with around 200, 500, and 1000 peers respectively. The convergence time is almost the same for P2P video systems with different sizes. The number of iterations doesn’t increase with the number of peers in the P2P live video system.

![Message overhead per peer](image)

Fig. 11: Control message overhead per peer

During the execution of our semi-distributed algorithm, peers also need to exchange control messages with the coordinator. To evaluate the incurred message overhead for systems in different scales, we adopt a new metric called control message overhead, which is defined as the ratio between the control traffic volume and the video traffic volume. The control message overhead incurred by our algorithm is shown in Fig. 11. We measured the control message overhead of a system with 200, 500, and 1000 peers respectively. It can be found that the message overhead per peer is pretty low (less than 0.1%) and doesn’t increase when the number of peers increases. It implies that our algorithm is a lightweight solution that can scale well with the system size.

VI. CONCLUSION

Fairness is an important problem in the P2P live video systems. In this paper, we developed a simple model to analyze the fairness problem in P2P live video systems. By modeling and analysis, we can derive the fairness and performance metrics in the optimal points and identify the tradeoff between fairness and performance. We also proposed a distributed algorithm to improve fairness level in the P2P live video system. In our current analysis, we only consider single-layer video and the results show that it is hard to achieve a very high level of fairness for the high-definition video streaming. In our future work, we plan to investigate the fairness problem in layered video streaming systems [39]. With layered video coding, the video stream is divided into multiple substreams with different importance. The fairness in the P2P streaming system can be significantly improved by letting a peer's upload rate be commensurate with the number of downloaded substreams. It is also of interest to study how to better integrate our algorithm with existing incentive mechanisms.

APPENDIX

A. Proof of Theorem III.1

Proof: By Little’s law, the number of k-type peers in the system under steady state is given by $N_k = \lambda_k T = p_k \lambda T$, where $T$ is the average peer sojourn time in the system. Thus, the total number of peers is given by $N = \sum_{k=1}^{K} N_k = \lambda T$. The total upload capacity of peers is given by $C = u_s + \sum_{k=1}^{K} u_k N_k = u_s + \lambda T \sum_{k=1}^{K} p_k u_k$. Considering the constraint of server upload rate $u_s$, the maximum achievable streaming rate can be derived as $R_{perf} = \min\{u_s, C/N\} = \min\{u_s, \frac{C}{\lambda T} + \sum_{k=1}^{K} p_k u_k\}$. $R_{perf}$ is defined as the maximum streaming rate that the system can sustain and universal streaming is achieved among all peers. $R_{perf}$ is bounded by the server upload rate $u_s$ and the average upload bandwidth allocated for each peer, which equals to the total upload capacity $C$ divided by the total number of peers $N$.

When $u_s > U_K$ and $N$ is large enough, the above equation can be rewritten as $R_{perf} = \frac{u_s}{\lambda T} + \sum_{k=1}^{K} p_k u_k$.

As $\frac{1}{\lambda T}$ is a fixed value under steady state, to optimize the performance, we only need to optimize the second term and solve the following linear program:

Maximize $\sum_{k=1}^{K} p_k u_k$

subject to $0 \leq u_k \leq U_k, k = 1, \cdots, K$;

The optimal solution is $u_k = U_k, k = 1, \cdots, K$. It means that, when every peer uploads at its maximum upload capacity, we can achieve the optimal universal streaming rate. Thus we have the maximum streaming rate as $R_{perf} = \frac{u_s}{\lambda T} + \sum_{k=1}^{K} p_k U_k$. The fairness index at that point can also be derived accordingly by $F_{perf} = \frac{(\sum_{k=1}^{K} p_k U_k)^2}{\sum_{k=1}^{K} p_k U_k^2}$.

B. Proof of Theorem III.2

Proof: According to the optimization theory, the fairness index is maximized when all the peers have the same upload rate, i.e., $u_1 = u_2 = \cdots = u_K = u$. At that time, $F_{fair} = 1$. Accordingly, the maximum achievable streaming rate is given by $R_{fair} = \frac{u}{\lambda T} + U_1$, as $u \leq \min\{U_1, U_2, \ldots, U_K\}$.

C. Proof of Theorem III.3

Proof: By KKT conditions, we have the following equations:

\[
\begin{align*}
\text{KKT 1:} & \quad \frac{\partial L}{\partial u_k} = 0 \\
\text{KKT 2:} & \quad \alpha_k g_1^{(k)}(u_k) = 0, \text{ and } \beta_k g_2^{(k)}(u_k) = 0 \\
\text{KKT 3:} & \quad \alpha_k \geq 0 \text{ and } \beta_k \geq 0 \\
\text{KKT 4:} & \quad h(u) = 0
\end{align*}
\]

To solve the above equations, we can consider three cases: Case I: In this case, all the peers upload less than their capacity but greater than zero, namely, $0 < u_k < U_k, \forall k$. By KKT 2,
we have $\beta_k = 0$ and $\alpha_k = 0$. From KKT 1, we have $u_k = \frac{r}{N^2}$. It means that the peers whose rate satisfies $0 < u_k < U_k$ have the same upload rate. If all peers have the same upload rate, $\sum_{k=1}^{K} p_k u_k = r' \Rightarrow u_k = r'(k = 1, \ldots, K)$. As $u_k < U_k$, we have $r' \leq \min(U_k, k = 1, \ldots, K) = U_1$, which equals to $r < U_1 + \frac{r}{N^2} \leq R_{fair}$. It contradicts with our assumption $R_{fair} < r < R_{perf}$. Thus, Case I doesn’t hold and only a portion of peers satisfy $0 < u_k < U_k$.

**Case II:** In this case, all the peers upload at their full capacity, namely, $u_k = U_k (k = 1, \ldots, K)$. It corresponds to the maximum performance case. We have $r = R_{perf} = U_1 + \sum_{k=1}^{K} p_k U_k$. Case II also doesn’t hold as it contradicts with our assumption $r < R_{perf}$.

**Case III:** In this case, some peers upload less than their capacity and other peers upload at full capacity. For $k$-type peers that satisfy $0 < u_k < U_k$, we have $u_k = \frac{r}{N^2}$. For $j$-type peers that satisfy $u_j = U_j$, we can obtain $p_j (2U_j + \nu) + \beta_j = 0$. From $\beta_j \geq 0$, we can derive $u_j = U_j \leq \frac{\nu}{N^2}$. The above results have the following implications: (1) It means that, if a peer uploads at its full rate $U_j$, the rate is lower than other peers that do not upload at the full rate $0 < u_k < U_k$ and $u_k = \frac{r}{N^2}$. (2) There exists a point $0 < m < K$ that peers whose type is no greater than $m$ upload at their full rate; for other peers whose type is greater than $m$, they have the same upload rate $u$. The threshold $m$ satisfies that $\sup\{ j | U_j \leq r' \leq U_{j+1} \} \leq m$ and $U_m \leq u \leq U_{m+1}$. As $\sum_{k=1}^{K} p_k u_k = r'$, we have $\sum_{k=1}^{m} p_k U_k + \sum_{k=m+1}^{K} p_k U_{m+1} \leq r' \leq \sum_{k=1}^{m} p_k U_k + \sum_{j=m+1}^{K} p_j U_{m+1}$. We can determine the value $m$ and $u$ that satisfy the above inequality and minimize $\sum_{k=1}^{m} p_k U_k^2$. At that time point, the fairness index is given by:

$$F = \frac{(r - \frac{u}{N^2})^2}{\sum_{k=1}^{m} p_k U_k^2 + \sum_{k=m+1}^{K} p_k U_k^2}.$$

### References


