IMPROVING THE STREAMING CAPACITY IN P2P VOD SYSTEMS WITH HELPERS

Yifeng He and Ling Guan

Department of Electrical and Computer Engineering, Ryerson University, Toronto, Ontario, Canada

ABSTRACT

Peer-to-Peer (P2P) Video-on-Demand (VoD) is a promising solution to provide video service to a large number of users. Streaming capacity in a P2P VoD system is defined as the maximal streaming rate that every user can receive. Due to the upload bottleneck, the streaming capacity in the P2P VoD system is limited. In this paper, we introduce helpers in the P2P VoD system and then optimize the helper resources to improve the streaming capacity. Specifically, we first optimize the helper assignment using a greedy algorithm. Then we develop a proximal distributed algorithm to maximize the streaming capacity by optimizing the link rates. Through simulations, we demonstrate that the P2P VoD system with optimized helpers can obtain a much higher streaming capacity compared to the P2P VoD system without any helper or the one with randomly assigned helpers.

Index Terms— Peer-to-peer video on demand, streaming capacity, helper, optimization

1. INTRODUCTION

With the increase of the link capacity offered to Internet users, Video on Demand (VoD) services are becoming popular. However, providing VoD to a large number of concurrent users requires a significant amount of bandwidth, which causes a bottleneck at a VoD server due to lack of scalability. It is appealing to apply Peer-to-Peer (P2P) technology into VoD applications to relieve the server upload burden by taking advantage of the uplink capability of the peers. The VoD system that combines the server and the peers is called the P2P VoD system.

In P2P VoD systems, users would like to watch the video at a high quality. Naturally, we are interested in the following question: what is the upper bound of the streaming rate in a P2P VoD system? In [1], streaming capacity is defined as the maximum supported streaming rate that can be received by every receiver. In our previous work [2], we found that the streaming capacity in a P2P VoD system is limited by the upload capacity. In order to improve the streaming capacity, we introduce helpers in a P2P VoD system. A helper is a node who is not interested in watching the video but is willing to use its idle bandwidth to download the video at a small rate and then forward the received rate to other peers.

Helpers have been recently proposed in P2P systems. The download performance of peers in a BitTorrent-like system can be improved by using the spare upload capacities of the helpers [3]. The streaming capacity in tree-based P2P live streaming systems with helpers was investigated in [1], where an approximation algorithm was developed to compute the streaming capacity. We studied the streaming capacity in the P2P VoD system without any helper in [2]. In the P2P VoD system with a predetermined streaming rate, the helpers can offload the server burden [4].

Different from the previous work [1, 2, 4], this paper focuses on the optimal utilization of helper resources to improve the streaming capacity in a P2P VoD system. Our contributions are as follows. First, we propose a greedy algorithm to optimally assign each helper to serve a segment. Second, we develop a proximal distributed algorithm to maximize the streaming capacity by optimally allocating the link rates.

2. SYSTEM OVERVIEW

A peer has to receive the video at the playback rate $r$ in order to decode and playback the video, while a helper only needs to receive a rate much smaller than the playback rate $r$ because it is not interested in watching the video. There are three roles in a P2P VoD system with helpers: 1) the server, 2) the peers, and 3) the helpers. We use the terminology, a node, to represent any entity in the system. Hence a node can represent a server, a peer, or a helper. The set of the nodes is denoted by $N$, among which there are three subsets: 1) the server subset, denoted by $S$; 2) the peer subset, denoted by $P$; and 3) the helper subset, denoted by $H$. Therefore we have $N = S + P + H$.

In a P2P VoD system, the peers organize themselves into an overlay based on the playback progress such that the peer with an earlier playback progress forwards the buffered content to the one with a late playback progress. When the peer cannot get sufficient supply from its supplying peers, it can ask the server to supply the remaining rate. To overcome the bottleneck of the upload bandwidth, we are motivated to utilize the extra bandwidth of the helpers who are not interested in the video but are willing to contribute their upload
bandwidth. A incentive mechanism should be established in the P2P VoD systems to encourage the contribution of the helpers. The P2P VoD system with helpers is illustrated in Fig. 1.

In order to reduce the overhead on the helper, we assume that a helper only downloads and forwards a segment at a time. At helper \( h ( h \in H ) \), the total incoming rate from the supplying peers is denoted by \( r_{h}^{in} \), which represents the amount of the peer upload bandwidth consumed by helper \( h \). The total outgoing rate to the requesting peers is denoted by \( r_{h}^{out} \), which represents the amount of the upload bandwidth contributed by helper \( h \). A helper can be thought of as a bandwidth amplifier in a P2P VoD system. A amplifier is characterized by the gain. Therefore we define a bandwidth gain \( \xi_{h} \) for helper \( h \), which is given by \( \xi_{h} = r_{h}^{out} / r_{h}^{in} \). A larger bandwidth gain indicates that the helper makes a greater contribution to the system.

The proposed optimized scheme consists of two steps: helper assignment and optimal rate allocation. The purpose of helper assignment is to determines which segment a helper should serve. Optimal rate allocation determines the link rates for each node in order to maximize the streaming capacity.

### 3. HELPER ASSIGNMENT

The video is evenly divided into multiple segments, denoted by a whole-segment set \( M \). At the current moment, the segments that are being watched is denoted by a watched-segment set \( W \). The set of the peers watching segment \( m ( \forall m \in W ) \) is denoted by a group set \( S_{m} \). The aggregate received bandwidth by the peers watching segment \( m \) is denoted by \( q_{m} \). We denote by \( z_{m} \) the Average Received Bandwidth (ARB) for the peers watching segment \( m \). The initial ARB for segment \( m \) without any helper is given by \( z_{m}^{(0)} = q_{m} / |S_{m}| \) where \( |S_{m}| \) denotes the number of the peers in the group set \( S_{m} \). Let \( y_{m} \) denote the upload bandwidth from the helpers allocated to the peers watching segment \( m ( \forall m \in W ) \). Then the ARB for segment \( m \) is given by \( z_{m} = (q_{m} + y_{m}) / |S_{m}| \). The aggregate upload bandwidth from the helpers is \( b_{tot}^{h} \). The summation of all the allocated bandwidth should be no larger than the aggregate upload bandwidth from the helpers, which is expressed by \( \sum_{m \in W} y_{m} \leq b_{tot}^{h} \). The minimal ARB among all the segments in the watched-segment set \( W \) is denoted by \( b = \min_{m \in W} \{ z_{m} \} = \min_{m \in W} \{ (q_{m} + y_{m}) / |S_{m}| \} \).

Since the streaming capacity is dependent on \( b \), our objective is to maximize \( b \) by optimizing \( y_{m} \) for each segment in the watched-segment set \( W \). Therefore the problem for allocating the helper upload bandwidth can be formulated as follows.

\[
\begin{align*}
\text{maximize}_{y} & \quad b \\
\text{subject to} & \quad b = \min_{m \in W} \{ (q_{m} + y_{m}) / |S_{m}| \}, \quad \forall m \in W, \\
& \sum_{m \in W} y_{m} \leq b_{tot}^{h}, \quad \forall m \in W. \\
\end{align*}
\]

The optimization problem (1) can be converted to the following equivalent form.

\[
\begin{align*}
\text{maximize}_{y,b} & \quad b \\
\text{subject to} & \quad (q_{m} + y_{m}) / |S_{m}| \geq b, \quad \forall m \in W, \\
& \sum_{m \in W} y_{m} \leq b_{tot}^{h}, \quad \forall m \in W. \\
\end{align*}
\]

The initial ARB for segment \( m \) is \( q_{m} / |S_{m}| \). By filling segment \( m \) with helper bandwidth \( y_{m} \), we can raise the ARB to \( z_{m} = (q_{m} + y_{m}) / |S_{m}| \). In order to maximize the minimal ARB, we need to fill the segment with the lowest ARB using the upload bandwidth from the helpers in an iterative way. In a P2P VoD system, helpers may arrive and leave dynamically. Therefore we propose an efficient greedy algorithm for helper assignment as shown in Table 1.

We require each helper to serve a segment at a time. Therefore the proposed greedy algorithm only provide a near-optimal solution to the optimization problem (2). After the helper has been assigned to serve a segment, it connects itself to a upstream peer who is buffering the segment, and also connects itself to the downstream peers who are requesting the segment.

### 4. OPTIMAL RATE ALLOCATION

The overlay of the P2P VoD system with helpers can be modeled as a directed graph \( G = (N, L) \), where \( N \) is the set of...
nodes and \( \mathbf{L} \) is the set of directed overlay links. The server is denoted by node 1. The relationship between a node and its outgoing links is represented with a matrix \( \mathbf{A}^+ \), whose elements are given by: \( a^+_{il} = 1 \) if link \( l \) is an outgoing link from node \( i \), and \( a^+_{il} = 0 \) for other cases. The relationship between a node and its incoming links is represented with a matrix \( \mathbf{A}^- \), whose elements are given by: \( a^-_{il} = 1 \) if link \( l \) is an incoming link into node \( i \), and \( a^-_{il} = 0 \) otherwise.

In order to distinguish the peers from the server and the helpers, we define a node-filtering element \( f_i \) as: \( f_i = 0 \) if \( i = 1 \) or \( i \in \mathbf{H} \), and \( f_i = 1 \) otherwise. In order to distinguish the links directly outgoing from the server from the other links, we define a link-filtering element \( g_l \) as: \( g_l = 1 \) if link \( l \) is a direct outgoing link from the server, and \( g_l = 0 \) otherwise.

The relationship between the outgoing link and the incoming link of a node is represented by a matrix \( \mathbf{C} \), whose elements are given by: \( c_{lm} = 1 \) if link \( m \) is an incoming link into the start node of link \( l \), and \( c_{lm} = 0 \) for other cases.

We assume that the bottleneck only appears at the uplink, and denote by \( O_i \) the upload capacity of node \( i \). The upload constraint at node \( i (\forall i \in \mathbf{N}) \) is given by \( \sum_{l \in \mathbf{L}} a^+_{il} x_l \leq O_i \) where \( x_l \) is the link rate at link \( l \). The upload constraint represents that the total outgoing rate from node \( i \) is no larger than its upload capacity \( O_i \). The streaming rate is denoted by \( r \). In a P2P VoD system with helpers, only the peers are required to receive the streaming rate \( r \). The helper only needs to receive a small portion of the streaming rate and then relay the received rate to other peers. At node \( i \) (\( i \neq 1 \)), each outgoing link from node \( i \) carries a rate no larger than the total incoming rate into it. Each direct outgoing link from the server carries a rate no larger than the streaming rate \( r \). This constraint is referred to as link-forwarding constraint, which can be formulated as \( x_l - \sum_{m \in \mathbf{L}} c_{lm} x_m \leq g_l r, \forall l \in \mathbf{L} \). Mathematically, the streaming capacity problem in the P2P VoD system with helpers can be formulated as follows.

\[
\begin{align*}
\text{maximize}_{(\mathbf{x}, r)} & \quad r \\
\text{subject to} & \quad f_i(\sum_{l \in \mathbf{L}} a^+_{il} x_l - r) = 0, & \forall i \in \mathbf{N}, \\
& \sum_{l \in \mathbf{L}} a^+_{il} x_l \leq O_i, & \forall i \in \mathbf{N}, \\
& x_l - \sum_{m \in \mathbf{L}} c_{lm} x_m \leq g_l r, & \forall l \in \mathbf{L}, \\
& x_l \geq 0, & \forall l \in \mathbf{L}.
\end{align*}
\]

The optimization problem (3) can be solved with a centralized algorithm. However, the centralized algorithm is non-scalable. In order to develop a distributed algorithm, we change the objective function in the optimization problem (3) to a strictly convex function as follows.

\[
\begin{align*}
\text{minimize}_{(\mathbf{x}, r)} & \quad -r + \epsilon r^2 + \epsilon \sum_{l \in \mathbf{L}} x_l^2 \\
\text{subject to} & \quad \text{the same constraints as in (3)},
\end{align*}
\]

where \( \epsilon (\epsilon > 0) \) is called a regularization factor. When \( \epsilon \) is small enough, the solution for the problem in (4) is arbitrarily close to the solution for the original streaming capacity problem (3).

The optimization problem (4) is a convex optimization problem [5], since it has a strictly convex objective function and the linear constraints. We can develop a distributed algorithm to solve it using the dual decomposition. We introduce dual variables \((u_i, v_i, \forall i \in \mathbf{N})\) and \((\lambda_l, \forall l \in \mathbf{L})\) to formulate the Lagrange dual problem corresponding to the primal problem (4) as below:

\[
\begin{align*}
\text{maximize}_{(\mathbf{u}, \mathbf{v}, \lambda)} & \quad G(\mathbf{u}, \mathbf{v}, \lambda) \\
\text{subject to} & \quad v_i \geq 0, & \forall i \in \mathbf{N}, \\
& \lambda_l \geq 0, & \forall l \in \mathbf{L}, \\
& G(\mathbf{u}, \mathbf{v}, \lambda) = \min_{(x \geq 0, r \geq 0)} (-r + \epsilon r^2 + \epsilon \sum_{l \in \mathbf{L}} x_l^2 + \sum_{i \in \mathbf{N}} u_i f_i(\sum_{l \in \mathbf{L}} a^+_{il} x_l - r) + \sum_{i \in \mathbf{N}} v_i(\sum_{l \in \mathbf{L}} a^-_{il} x_l - O_i) + \sum_{l \in \mathbf{L}} \lambda_l(x_l - \sum_{m \in \mathbf{L}} c_{lm} x_m - g_l r)).
\end{align*}
\]

where \( G(\mathbf{u}, \mathbf{v}, \lambda) = \min_{(x \geq 0, r \geq 0)} (-r + \epsilon r^2 + \epsilon \sum_{l \in \mathbf{L}} x_l^2 + \sum_{i \in \mathbf{N}} u_i f_i(\sum_{l \in \mathbf{L}} a^+_{il} x_l - r) + \sum_{i \in \mathbf{N}} v_i(\sum_{l \in \mathbf{L}} a^-_{il} x_l - O_i) + \sum_{l \in \mathbf{L}} \lambda_l(x_l - \sum_{m \in \mathbf{L}} c_{lm} x_m - g_l r)). \)

We use subgradient method [6] to solve the Lagrange dual problem (5). The dual variables \((u_i^{(k+1)}, v_i^{(k+1)}, \lambda_l^{(k+1)})\) at the \((k+1)\)th iteration are updated respectively by:

\[
\begin{align*}
u_i^{(k+1)} &= u_i^{(k)} - \theta^{(k)} f_i(r^{(k)} - \sum_{l \in \mathbf{L}} a^+_{il} x_l^{(k)}), \forall i \in \mathbf{N}, \\
\lambda_l^{(k+1)} &= \max\{0, \lambda_l^{(k)} - \theta^{(k)} (O_l - \sum_{i \in \mathbf{N}} a^-_{il} x_l^{(k)})\}, \forall l \in \mathbf{L}, \\
\theta^{(k)} &= \frac{\max(0, \lambda_l^{(k)} - \theta^{(k)} (\sum_{m \in \mathbf{L}} c_{lm} x_m + g_l r_l - x_l^{(k)}))}{\forall l \in \mathbf{L}, \text{where } \theta^{(k)}(\theta^{(k)} > 0) \text{ is the step size at the } k^{th} \text{ iteration}}.
\end{align*}
\]

The server is responsible for computing the optimal streaming rate. The streaming rate \( r^{(k)} \) at the \( k^{th} \) iteration is updated by \( r^{(k)} = \max\{0, 1 + \sum_{i \in \mathbf{N}} u_i^{(k)} f_i + \sum_{l \in \mathbf{L}} \lambda_l^{(k)} g_l\}/(2\epsilon) \). The link rates can be computed in parallel. At the \( k^{th} \) iteration, the link rate \( x_l^{(k)} \) at link \( l \) is updated by \( x_l^{(k)} = \max\{0, -\beta_l/2\epsilon\}, \forall l \in \mathbf{L} \), where \( \beta_l = \sum_{i \in \mathbf{N}} (u_i^{(k)} f_i a^+_{il} + v_i^{(k)} a^-_{il}) + \lambda_l^{(k)} - \sum_{m \in \mathbf{L}} \lambda_m^{(k)} c_{lm} \).

5. SIMULATIONS

In the simulations, the upload capacity of the server is set to 3 Mbps. There are two classes of peers: cable/DSL peers and Ethernet peers. Cable/DSL peers take 85% of the total population with download capacity uniformly distributed between 0.9 Mbps and 1.5 Mbps and upload capacity uniformly distributed between 0.3 Mbps and 0.6 Mbps. Ethernet peers take the remaining 15% of the total population with both upload and download capacities uniformly distributed between 1.5 Mbps and 3.0 Mbps. The number of the helpers is 10% of the number of the peers if not specified particular. The length of the video is 60 minutes, which is evenly divided into 60 segments. The playback time of each peer is uniformly distributed between 0 and the length of the video. In the proximal distributed optimization, the regularization factor is set to 0.02.

We compare the maximal streaming rate among four schemes: 1) Scheme 1: Without Helper (WH), in which there is no helper, and the link rates are optimized in a centralized way; 2) Scheme 2: Random Assignment and Centralized Rate
Optimization (RACRO), in which each helper is randomly assigned to serve a segment in the watched-segment set \( W \), and then the link rates are optimized by solving the optimization problem (3) with a centralized algorithm; 3) Scheme 3: Optimal Assignment and Proximal Distributed Rate Optimization (OAPDRO), in which each helper is optimally assigned with the proposed greedy algorithm, and then the link rates are optimized by solving the optimization problem (4) with the proposed distributed algorithm; and 4) Scheme 4: Optimal Assignment and Centralized Rate Optimization (OACRO), in which each helper is optimally assigned with the proposed greedy algorithm, and then the link rates are optimized by solving the optimization problem (3) with a centralized algorithm. Fig. 2 shows the comparison of the maximal streaming rate when the number of the peers is varied from 100 to 300. Due to random assignment, Scheme 2 (RACRO) improves the maximal streaming rate only 5.5% in average compared to the scheme without any helper. If the helper is optimally assigned, the improvement of the streaming rate is much significant. Scheme 3 (OAPDRO) and Scheme 4 (OACRO) improve the maximal streaming rate by 20.5% and 28.3% compared to the scheme without any helper, respectively. The proposed scheme, OAPDRO, performs the rate allocation in a distributed manner, which is suitable for P2P networks.

The convergence of the maximal streaming rate in a P2P VoD network with 200 peers is shown in Fig. 3(a). Each peer is connected to 10 upstream peers at most. After 455 iterations, the maximal streaming rate with the proposed OAPDRO scheme converges to 0.451 Mbps, very close to the optimal streaming rate 0.496 Mbps obtained from the centralized scheme (OACRO). Each helper in the P2P VoD network acts as a bandwidth amplifier. We study the bandwidth gains of these amplifiers in a P2P VoD network with 200 peers in Fig. 3(b). In the RACRO scheme where each helper is randomly assigned to a segment, only three helpers (helper 3, 5 and 20) amplify the upload bandwidth, while the other 17 helpers contribute zero bandwidth to the peers. On the other hand, the proposed OAPDRO scheme optimizes both the helper assignment and the rate allocation such that each of the helpers makes a contribution to the P2P system, with an average bandwidth gain of 4.85. In the proposed OAPDRO scheme, the average received rate at a helper is 0.089 Mbps, which indicates that the download burden on the helper is very light.

6. CONCLUSION

In this paper, we utilize the upload bandwidth of the helpers to improve the streaming capacity. Specifically, we optimally assign each helper to serve a segment with a greedy algorithm. Then we develop a proximal distributed algorithm to maximize the streaming capacity by optimizing the link rates. Through simulations, we demonstrate that the proposed optimized scheme can improve the streaming capacity significantly compared to the P2P VoD system without any helper or the P2P VoD system with randomly assigned helpers.

7. REFERENCES