

P2Pを用いたビデオストリーミング 配信システムの効率化に関する検討

池田 匡視^{†1} 岡崎 直宣^{†2} 朴 美娘^{†3}

近年, インターネットの広帯域化に伴い, ビデオストリーミング配信の需要が高まっている. クライアント・サーバ型ストリーミング配信では, 配信サーバに負荷が集中してしまうという問題がある. そこで, P2P ネットワーク上でストリーミング配信を行う P2P 型ストリーミング配信が注目されている. P2P 型ストリーミング配信では, 負荷をピア間に分散できる. しかし, P2P 型では配信サーバほど送信帯域を持っていないピアが配信を行うため, ユーザの待ち時間が問題となる. そこで本論文では, ユーザが次に再生する動画を予測し, あらかじめバッファリングしておくことで, 次の動画の待ち時間を減少させる手法を提案する. 本手法では, 各スーパーノードがユーザの再生履歴を記憶しておき, その情報を元にユーザが次に再生する動画の予測を行う. また, シミュレーションを行い, 提案手法の有用性を示す.

A Study on video streaming protocol on Peer-to-Peer systems

MASAMI IKEDA,^{†1} NAONOBU OKAZAKI^{†2}
and MIRANG PARK^{†3}

Recently, demand for video streaming delivery is rising with spread of broadband access. Load concentrates on a delivery server in the Client-server type streaming delivery. In this situation, P2P type streaming delivery attracts attention. However, in the P2P type streaming delivery, the waiting time for caching data is a problem. Because a peer having the bandwidth which is smaller than a delivery server having. In this paper, we propose a video streaming protocol on Peer-to-Peer systems to decrease the caching time. In our method, a peer predicts the video which user wants to play next and buffers it beforehand. We show the waiting time of the next video is reduced by simulation.

1. はじめに

近年, インターネットの広帯域化に伴い, マルチメディア配信の需要が高まっている. そのなかでもビデオストリーミング配信が注目されている. ビデオストリーミング配信 (以降ストリーミング配信) とは, 動画データの受信を行いながら同時に再生を行う方法で, 動画データのダウンロード終了まで待たずに再生を開始できることがその特徴である.

ストリーミング配信において, 配信を行うストリーミングサーバは, ファイルサイズが大きく時間軸の情報を持った動画データを配信し, さらにそのデータを帯域や送信先の数に合わせて送信量を調整しながら配信しなければならない. そのため, ファイルの一括ダウンロードを行うような一般的なサーバよりも処理が複雑になってしまい, サービス拡大時や安定した配信を行う為にコストが高くなる.

この問題を解決する手段として P2P(Peer to Peer) ネットワーク上でストリーミング配信を行う P2P 型ストリーミング配信が研究されている¹⁾⁻⁴⁾. P2P ネットワークとは, ネットワークに接続された一般ユーザのコンピュータ (ピア) 同士が直接通信を行う方式でサーバを必要としない. 従来のクライアント・サーバ型ではストリーミングサーバに集中していた配信負荷をピア間に分散できるため, 配信負荷を軽減できるという特長がある.

一般的にストリーミング配信において, 動画データはある再生時間間隔で分割して扱われる. その一つ一つをセグメントと呼ぶ. ストリーミング配信では, クライアントはセグメントの受信と同時に再生を行うが, 受信に再生が追いつかないように再生開始前にしばらく一定量のセグメントをバッファに蓄える (これをバッファリングと呼ぶ). そのため動画の再生を開始するまでに待たされてしまう. P2P 型ストリーミング配信では, ピアが配信も行うが, 一般ユーザのコンピュータであるピアは配信サーバほど送信帯域を持っていないことが多い. そのため, P2P 型ストリーミング配信ではクライアント・サーバ型に比べ, 待ち時間が長くなりやすい. さらに, クライアント・サーバ型では配信サーバが全ての動画を管理しているのに対し, P2P 型では各ピアが動画の所在を管理しているため, 検索に時間がかかってしまい, このことがユーザの待ち時間の増加につながる.

そこで本研究では, 再生開始までの待ち時間を短縮させる手法を提案する. 本手法では, 現在再生中の動画のバッファリング終了後, 次に再生する動画を予測し, それを予めバッ

^{†1} 宮崎大学大学院

^{†2} 宮崎大学

^{†3} 神奈川工科大学

ファリングすることで再生開始までの待ち時間を減らすアプローチをとる．さらに本研究では，ピア間の検索効率を向上させ，待ち時間を減らすための検索手法についても検討する．

2. 関連研究

関連研究として，ピアの帯域幅を考慮しデータ受信時の待ち時間を短縮するピア選択手法⁵⁾ や，不人気動画視聴ピアの帯域を有効活用する手法⁶⁾ がある．ここでは不人気動画視聴ピアの帯域を有効活用する手法である PEC(Popularity Based Efficient Caching) を説明する．

一般的な P2P 型ストリーミング配信では，ピアは動画のキャッシュを FIFO (First In First Out) で管理している．そのため，不人気動画を視聴した場合，不人気動画がキャッシュに残り，人気動画がキャッシュから溢れてしまうことがある．不人気動画では配信要求がほとんど来ないため，ピアの送信帯域を有効に利用できない．PEC では，この不人気動画視聴ピアの送信帯域を有効活用することで，配信サーバのトラフィック量を低減させている．

図 1 に，FIFO を用いた場合と，PEC でのセグメント送信者の例を示す．FIFO では人気動画視聴ピアである A,B,C はキャッシュに人気動画のセグメントを持つため，他ピアにセグメントを送信できる．しかし，不人気動画視聴ピアである D は，キャッシュに不人気動画のセグメントしか持っていないため，他ピアにセグメントを送信できない．PEC では，不人気動画視聴ピアである D もキャッシュに人気動画のセグメントを持つ．そのため，人気動画視聴ピアである A,B,C に加え，ピア D も他ピアにセグメントを送信できる．FIFO に比べ不人気動画視聴ピアである D もセグメントを送信できる分，配信サーバの負荷が低くなる．

PEC では，人気動画のセグメントは，FIFO でキャッシュに残し，不人気動画のセグメントはキャッシュに残さない．図 2 にピアによるセグメントのキャッシュ保持手順を示す．ピアはまず，セグメント探索要求を tracker サーバに送信する．tracker サーバは，セグメント探索要求を受信後，セグメント探索結果を返す．セグメント探索結果では，そのセグメントをどの相手からダウンロードすべきかと，視聴後にキャッシュに保持すべきかを伝える．そしてピアは tracker サーバから指定された相手にセグメント要求を送信する．ピアはセグメント視聴後，tracker サーバからの判断に従い，キャッシュに保持または破棄する．以上により，ピアは tracker サーバの判断に従ってセグメントをキャッシュに保持する．

PEC の欠点として，待ち時間を考慮していない点が挙げられる．ストリーミング配信のようリアルタイムにビデオを再生する環境では，待ち時間を考慮すべきである．本研究で

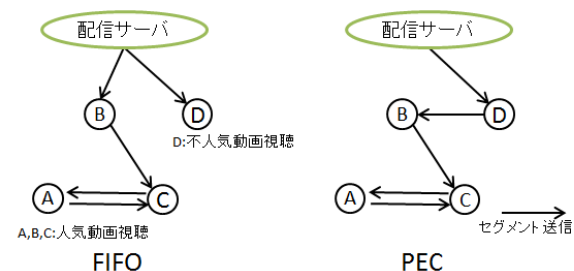


図 1 PEC におけるセグメント送信者の例
Fig. 1 Example of the segment sender on PEC.

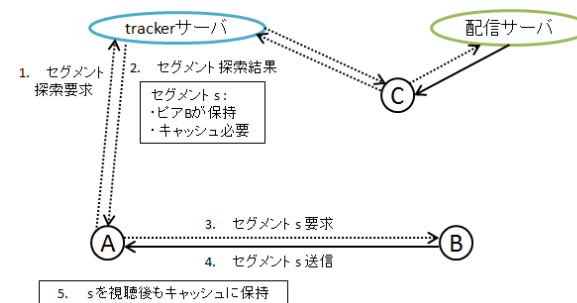


図 2 PEC におけるセグメントのキャッシュ保持の流れ
Fig. 2 Flow of the cash maintenance of the segment on PEC.

は，PEC におけるピアの送信帯域を有効活用する方法を取り入れ，さらに再生開始までの待ち時間を短縮する手法を提案する．

3. 提案手法

ストリーミング配信では，動画再生開始前に予め余分にセグメントをバッファに溜めておき，バッファリングに再生が追いつかないようにしている．これにより，再生中に途切れることは起きにくくなるが，再生開始するまでに長時間待たなければならない．

本研究では，この再生開始までの待ち時間を短縮することを考える．再生開始時に余分にバッファリングするため，動画の後半はバッファリングが終わり，何も通信していない場合が多い．そこで，この動画の後半にユーザが次に再生する動画をバッファリングしておけ

ば、次の動画に関しては再生開始までの待ち時間を大幅に短縮できる。

提案手法では、P2P 型ストリーミング配信において、現在の動画再生中に次の動画の先読みバッファリングを行う。この先読みバッファリングを行う為に、自動で動画の再生順序の予測を行う。動画の再生順序の予測は、子ノードが動画再生終了後、次の動画の再生開始時に、スーパーノードに動画の再生順序を含んだパケットを送信することで行う。このパケットを元にスーパーノードが動画の再生順序を予測する。この方法であれば、サービス提供者があらかじめ動画の順序関係を作成する必要がなく、サービス開始後も手を加える必要がない。スーパーノードについては 3.1 で述べる。

提案手法ではさらに、動画検索に割かれる時間を短縮する為、動画の検索効率を向上させる。検索効率を上げる方法として、発見した動画に関する応答パケットを転送した際に、スーパーノードはパケットの中身を参照する。応答パケットを参照することで、スーパーノードは自身の配下以外の子ノードのキャッシュ保持状況を知ることができる。次に検索要求が来た際にその情報を利用することで、他スーパーノードに要求を出す必要がなく即返答できる。さらに子ノードの送信帯域を有効に使うために、視聴後にその動画をキャッシュに残すべきかの判断も、スーパーノード間の検索時に行う。

3.1 対象とする P2P ネットワークアーキテクチャ

P2P ネットワークは、情報の検索方法によって、ピア型、ハイブリッド型、スーパーノード型に分類できる⁷⁾。提案手法では、ある程度情報をまとめて管理し、かつ耐障害性も持つスーパーノード型 P2P ネットワークを用いる。この対象とする P2P ネットワークアーキテクチャは、スーパーノード型 Gnutella⁸⁾ を参考にした。本手法の構成図を図 3 に示す。対象とする P2P ネットワークは、処理能力の高いスーパーノードと、処理能力の低い子ノードからなる。一台のスーパーノードに対して、複数の子ノードが接続する。子ノードは一台のスーパーノードとのみ接続しており、そのスーパーノードを親スーパーノードと呼ぶ。

動画の検索はスーパーノード間で検索パケットをフラッディングして行い、実際のストリーミング配信は子ノード同士で行う。スーパーノードは、自身の配下の子ノードがどの動画をキャッシュに持っているかをリストに持つ。このリストを以降「子ノードリスト」と呼ぶ。検索要求を受信すると子ノードリストを参照し、リスト内に要求動画をもつノードがあれば要求元ノードへ応答を返す。リスト内に要求された動画がなければ隣接するスーパーノードへ検索パケットを送信する。一定時間探しても要求動画が見つからなければ、配信サーバに配信してもらう。

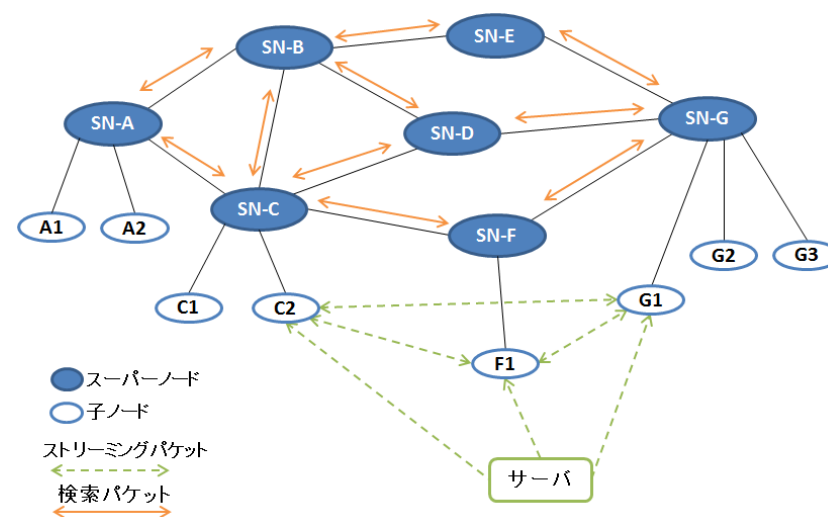


図 3 対象とする P2P ネットワークアーキテクチャ
Fig. 3 Target P2P network architecture.

3.2 システム全体の動作

3.2.1 動画検索の手順

提案手法における動画検索の手順を図 4 に示す。この図は、スーパーノード SN₁ の一番目の子ノード C1.1 が、動画の要求を送信し、実際にストリーミング配信が行われるまでの流れを表している。図中の n は、スーパーノード間での検索パケットのホップ数を表す。子ノードはこれから再生する動画の要求を自身の親スーパーノードに送信する (図 4(1) 参照)。動画の要求を受け取ったスーパーノードは、人気度測定に用いる「要求回数リスト」の更新を行う (図 4(2))。次に、自身の持つ「子ノードリスト」と「キャッシュノードリスト」内に要求動画がないか調べる (図 4(3))。キャッシュノードリストとは自身の配下以外の子ノードのキャッシュ状況を記録したリストである。どちらのリストにも要求動画が無い場合、スーパーノード内人気度判断 (図 4(4))、検索パケットの更新 (図 4(5)) を行い、隣接するスーパーノードへ検索パケットを転送する (図 4(6))。

スーパーノード内人気度判断 (図 4(4)) では、スーパーノード内において人気であると判断した場合、検索パケットに含まれる「人気度カウンタ」を 1 増やす。この人気度カウンタ

は、動画を発見したスーパーノードがキャッシュ判断を行うために用いる。検索パケットの更新 (図 4(5)) では、検索パケット内の転送回数を 1 増やし、経路に自身のアドレスを追加する。

転送された検索パケットを受信したスーパーノードは、自身の子ノードリストとキャッシュノードリスト内に要求動画がないか調べる (図 4(7))。存在しない場合、スーパーノード内人気度判断後、検索パケットの更新を行う。

この動作を繰り返し行い、要求動画を発見したスーパーノードは、要求動画のキャッシュ判断を行う (図 4(8))。キャッシュ判断とは子ノードが要求動画の再生終了後、その動画をキャッシュに残すべきかを定めることである。ここで、キャッシュ判断に必要なキャッシュ判断値を $Cache_{threshold}$ 、人気度カウンタを P_{count} 、転送回数を T とすると、以下の式でキャッシュ判断値を求めることができる。

$$Cache_{threshold} = P_{count}/T$$

このキャッシュ判断値は、複数のスーパーノード内での要求動画の人気度を数値化したものである。キャッシュ判断値が閾値以上ならば、多くのスーパーノード内で人気があることを示しており、キャッシュに残すべきであると判断する。キャッシュ判断により、子ノードは人気動画のみをキャッシュに残すことになり、子ノードの送信帯域を有効活用できる。

キャッシュ判断後、スーパーノードは、キャッシュ判断結果と自身が持つ「ネクスト動画リスト」を含めた応答パケットを作成し (図 4(9))、検索時に辿ってきた経路を遡り要求元ノードへと送信する (図 4(10))。

応答パケット転送経路上のスーパーノードは、応答パケット内の要求元ノードアドレスと、要求動画名をキャッシュノードリストに加える (図 4(11))。キャッシュノードリストを作成することで、応答パケット転送経路上のスーパーノードは、他スーパーノード配下の子ノードのキャッシュ状況を知ることができる。このキャッシュノードリストを利用することで、配下の子ノード内に要求動画が無い場合に、他スーパーノードへ検索要求を出すことなく要求元子ノードに応答を返すことができる。これにより、検索効率が上がリ、検索にかかる時間を短縮することができる。

応答パケットを受け取った子ノード (図 4(13)) は、応答パケットに含まれる配信ノードのアドレスを元に、配信ノードへと配信要求を送信する (図 4(14))。配信要求を受け取ったノードは要求元ノードへストリーミング配信を行う (図 4(15))。

3.2.2 動画再生の手順

子ノードの動画再生時の手順を図 5 に示す。子ノードは、現在再生している動画のパッ

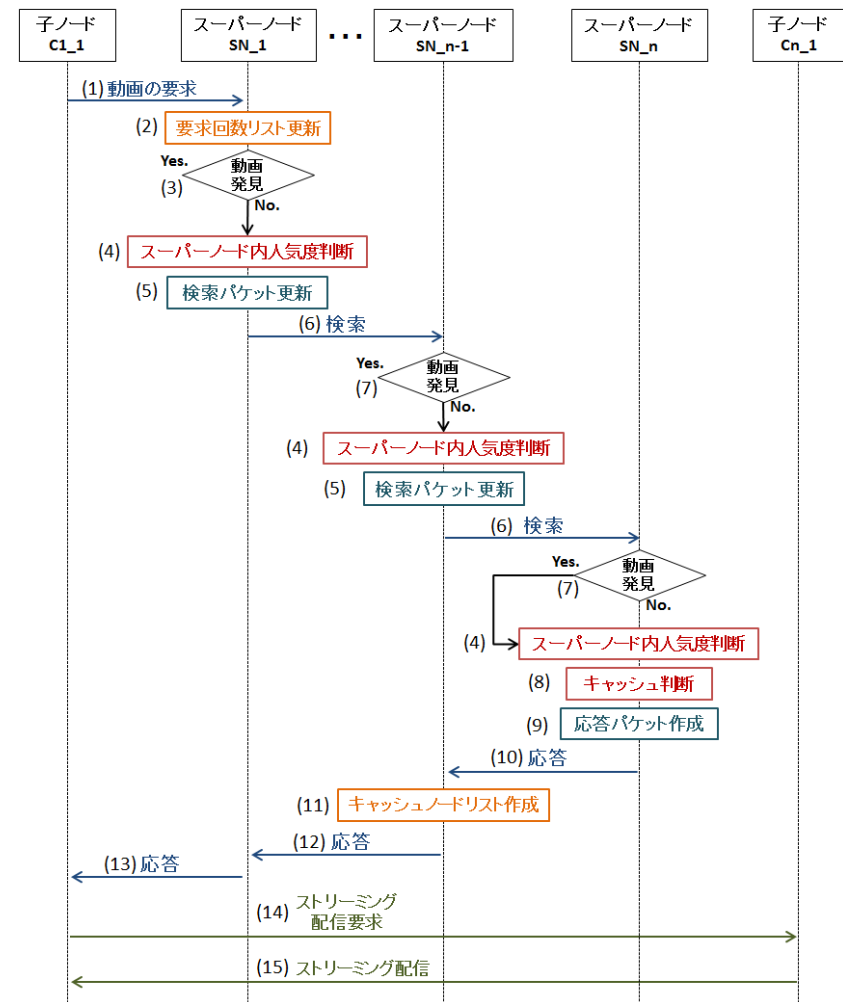


図 4 動画検索の手順
Fig. 4 Flow of the video search.

ファリング終了後(図 5(15)), 応答パケットに含まれる「ネクスト動画リスト」を元にして, ユーザに次に再生する動画を選択してもらう(図 5(16)). ネクスト動画リストとは, ある動画に対して, その動画の次に再生した動画名と何回その順序で再生されたのかを記録したものであり, スーパーノードが管理している. 詳しくは「3.4.4 ネクスト動画リスト更新パケット受信時」で説明する.

ユーザが次動画を選択した場合, 現在の動画再生中に親スーパーノードに対し, 選択された次動画の要求を送信する(図 5(17)). 次動画の検索(図 5(18))は, 図 4 の(2)-(12)と同じ動作で行う. 次動画の応答を受信後(図 5(19)), 次の動画のバッファリングを開始する(図 5(20)). 現在の動画再生中に次の動画のバッファリングを行うことで, ユーザは次の動画再生開始時に待たされることが大幅に少なくなる.

次動画の再生開始直後に, スーパーノードが持つネクスト動画リストを更新するために, 更新パケットを送信する. この更新パケットには現在再生が終了した動画の動画名と次動画名が含まれる. この更新パケットを, 次の二つのスーパーノードに送信する(図 5(21),(22)).

- 自身の親スーパーノード
- 現在再生が終了した動画を配信してもらった子ノードの親スーパーノード

3.3 子ノードの動作

3.3.1 動画要求時

子ノードは親スーパーノードにこれから再生する動画の要求を送信する. その後スーパーノードから応答パケットを受信する. 一定時間経っても応答がない場合は配信サーバに配信を要求する. スーパーノードからの応答には「ネクスト動画リスト」が含まれている. これはユーザが現在再生中の動画の次に再生する可能性のある動画のリストである. ユーザがこのリストから次動画を選択すると, 現在の動画再生中に自動で次動画のバッファリングを行う. これにより, 次動画再生開始時に待たされることは少なくなる.

3.3.2 動画再生終了時

子ノードは動画の再生が終了すると, スーパーノードが管理するネクスト動画リスト更新のため, スーパーノードに更新パケットを送信する. この更新パケットは現在再生が終了した動画名と次に再生する動画名からなる. この更新パケットを現在まで再生していた動画配信ノードの親スーパーノードと, 自身の親スーパーノードに送信する.

3.4 スーパーノードの動作

3.4.1 子ノードからの動画要求受信時

子ノードからの動画要求を受信したスーパーノードは, まず「要求回数リスト」の更新を

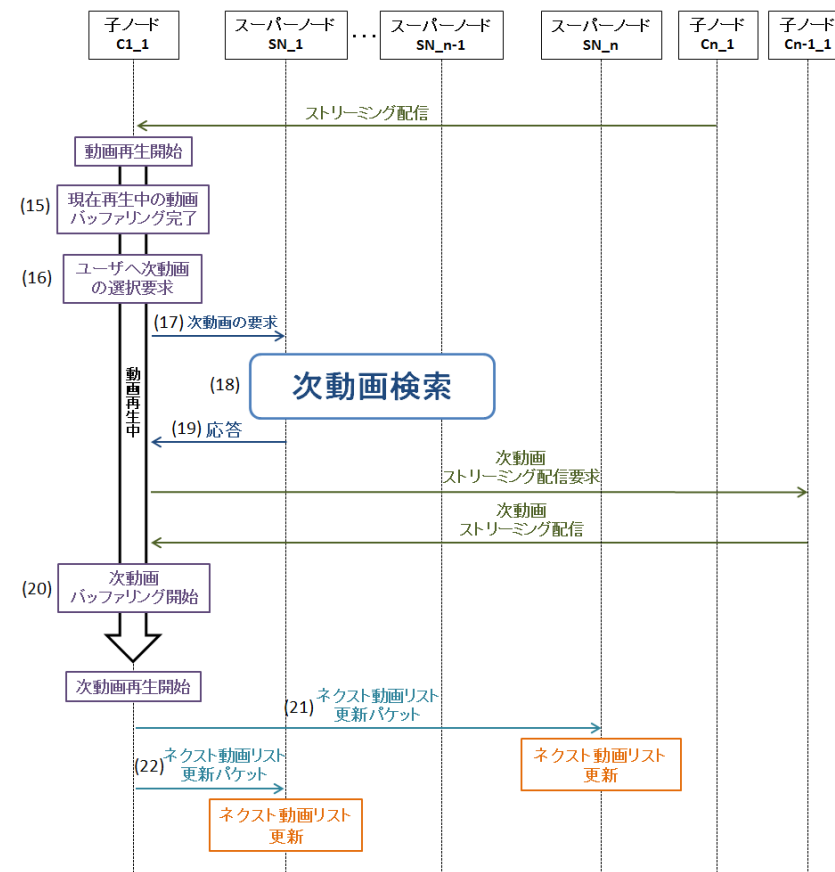


図 5 動画再生中の手順
Fig. 5 Flow during video playback.

行う。要求回数リストとは、ある動画が何回配下の子ノードに要求されたのかを記録したものであり、要求回数でソートされている。この要求回数リストは、「3.4.2 検索パケット転送時」でのスーパーノード内人気度判断で使用する。

次にスーパーノードは、自身の配下の子ノード内に要求動画をキャッシュに持っているノードがないかを「子ノードリスト」を用いて探す。無い場合、次に「キャッシュノードリスト」を探す。キャッシュノードリストとは、自身の配下以外の子ノードのキャッシュ状況を記録したものである。作成方法は「3.4.3 要求動画発見時」で説明する。キャッシュノードリストにも無い場合、隣接するスーパーノードに検索パケットを転送する。

3.4.2 検索パケット転送時

スーパーノードは検索パケット転送時、その動画が自身の配下の子ノードの中で人気があるかどうかを調べる。人気判断は、要求回数リストを用いて行う。要求回数リストで上位に位置する動画ならば配下の子ノードの間で人気がある動画であると判断する。その場合、検索パケット内の人気度カウンタを1増やす。この人気度カウンタは、「3.4.3 要求動画発見時」において、キャッシュ判断を行う際に使用する。

3.4.3 要求動画発見時

子ノードリストまたはキャッシュノードリストに要求動画があった場合、スーパーノードは要求元子ノードに向けて応答を返す。この時、子ノードが要求動画再生終了後キャッシュに残すべきかを決定する。このキャッシュ判断は検索パケット内の人気度カウンタを転送回数で除算し、その値が閾値以上ならば多くのスーパーノード間で人気がある動画だと判断し、キャッシュに残すべきだとする。このキャッシュすべきかを表す真偽値と、要求動画に対応するネクスト動画リストを応答パケットに含め、検索パケット転送経路を辿って要求元ノードへ送信する。応答パケット転送経路上のスーパーノードは応答パケットを転送する際、要求元子ノードのアドレスと、動画名をキャッシュノードリストに格納する。これにより、次回要求が来た場合すぐに返答できる。

3.4.4 ネクスト動画リスト更新パケット受信時

ネクスト動画リスト更新パケットを受信したスーパーノードは、自身の持つネクスト動画リスト内の対応する箇所の回数を1増やす。ネクスト動画リストの要素は以下の3つである。

- 動画名
- 次動画名
- 回数

表 1 シミュレーションパラメータ
Table 1 Simulation parameters.

スーパーノード数	10
子ノード数	100
ピア送信帯域	1Mbps
ピア受信帯域	2Mbps
動画ビットレート	320kbps
動画の長さ	180sec
セグメントサイズ	1sec
再生開始に必要なセグメント数	18
シミュレーション時間	180000sec

このリストは、「動画名」の次に「次動画名」を再生した「回数」を記録していく。各「動画名」の中に複数の「次動画名」があり、「回数」でソートされている。

この「次動画名」の上位複数を「3.4.3 要求動画発見時」の応答パケットに含めることで、応答パケットを受信した子ノードは、次動画をあらかじめバッファリングできる。

4. 評価

本章では提案手法の有用性を示すためシミュレーション実験を行う。本実験ではネットワークシミュレータ NS2(Network Simulator Version2)⁹⁾ の ns-2.34 を使用し、P2P 型ストリーミング配信機能全般と、提案手法の次動画の先読みバッファリング機能のみを実装し、シミュレーションを行った。

4.1 シミュレーション実験

実験は、提案手法と単純手法について行った。単純手法とは、次動画の先読みバッファリングを行わない方法である。ユーザの動画選択方法として、確率 p で現在再生している動画 ID+1 の動画を選択し、確率 $(1-p)$ で Zipf の法則に従い動画を選択するようにした。なお、実際の動画共有サービスでの動画の人気度が Zipf の法則にほぼ従うことが示されている⁶⁾。提案手法では、動画再生中に次動画を先読みしてバッファリングするが、ネクスト動画リストにユーザが再生したい動画が存在した場合のみ、先読みを行うことにした。実験方法は、各ノード毎に待ち時間を計測し、動画 10 回再生毎に待ち時間の平均を求めた。表 1 にシミュレーション条件を示す。

4.2 結果

実験結果を図 6 に示す。図 6 は動画再生開始までの待ち時間の平均である。動画再生中の中断は単純手法、提案手法共に発生しなかった。単純手法では再生回数に関係なく約 6 秒

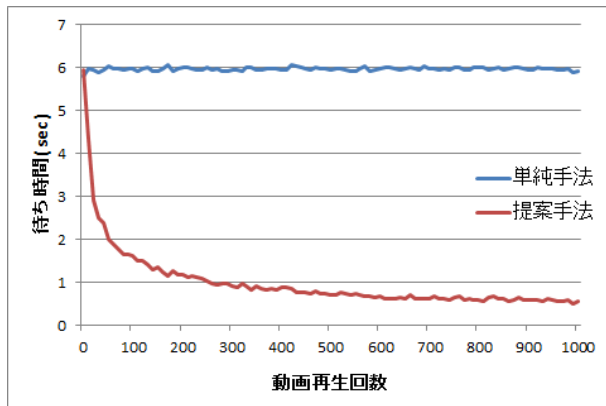


図 6 再生開始までの待ち時間
Fig. 6 Waiting time to start of playing video.

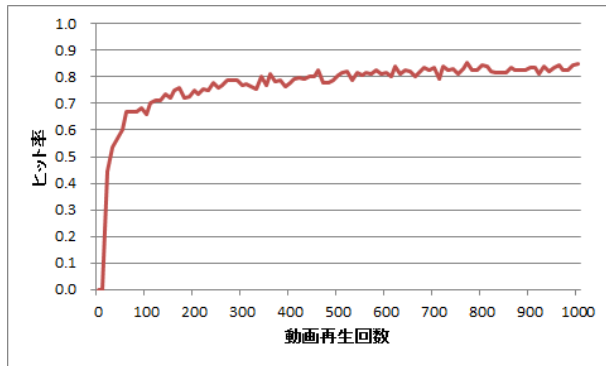


図 7 ネクスト動画リスト内に再生したい動画が存在した割合
Fig. 7 Hit ratio in the recommended next video list.

待ち時間が発生している。一方、提案手法では再生回数が増えるに従い待ち時間が減少している。図 7 にネクスト動画リストのヒット率を示す。ヒット率とは、ネクスト動画リスト内にユーザが再生したい動画が存在する割合である。

4.3 考察

以下の節で、待ち時間と提案手法におけるオーバーヘッドについての考察を行う。

4.3.1 待ち時間

図 6 より、提案手法は単純手法に比べ再生開始までの待ち時間を大幅に短縮できていることがわかる。提案手法では、動画再生回数が増加するに従い待ち時間が減少している。これは、時間経過と共にスーパーノードが持つネクスト動画リストが充実し、応答パケットに含まれるネクスト動画リスト内にユーザが再生したい動画が存在する回数が増えたためである。これは、図 7 において、時間経過と共にヒット率が上がっていることから確かめられる。

図 6 より、動画再生回数が 100 回を超えたあたりから安定して待ち時間が短縮されていることが分かる。本論文でのシミュレーション環境の場合、一日に一人のユーザが 5 回動画を再生したとすると、安定して待ち時間が短縮される段階までネクスト動画リストが充実するのに、サービス開始時から 20 日かかることになる。ユーザ数が増えればさらに短期間で安定することになるため、十分実用的であると考えられる。

4.3.2 ネットワークトラフィック

提案手法がネットワークトラフィックに与える負荷についての考察を行う。提案手法において、増加するパケットはネクスト動画リスト更新パケットのみである。ネクスト動画リスト更新パケットは、一回の動画再生毎に送信される。ネクスト動画リスト更新パケットには、前動画名と次動画名が含まれる。それぞれ最大文字数を 120 文字で 2 バイト文字だと仮定すると、ネクスト動画リスト更新パケットサイズは、ヘッダを含めて 500 バイトである。一つの動画サイズは、320kbps で 180 秒の場合、約 7M バイトである。これより、ネクスト動画リスト更新パケットは、一つの動画サイズの約 1/7000 である。よって、動画サイズに比べ、ネクスト動画リスト更新パケットは十分に小さいので、ネクスト動画リスト更新パケットによるネットワークへの負荷は無視できるものと考えられる。

4.3.3 スーパーノードへの負荷

次に、スーパーノードへの負荷についての考察を行う。提案手法において、スーパーノードで必要となる記憶領域は以下の 3 つである。

- 要求回数リストのための記憶領域
- キャッシュノードリストのための記憶領域
- ネクスト動画リストのための記憶領域

ここで、要求回数リストは、全動画数を一万、動画名の最大文字数を 120 文字の 2 バイ

ト文字とし、回数を4バイトに記録すると、必要となる記憶領域は最大で

$$(120 \times 2 + 4) \times 10^4 \approx 2.5MB$$

となる。

また、キャッシュノードリストは、キャッシュ保持時間が過ぎると対応する動画の項目を削除するため、要素数は最大でも全子ノード数となる。よって、最大子ノード数を一万、アドレスのサイズを4バイト、動画名を120文字の2バイト文字、動画の再生時間を8バイトに記録すると、必要となる記憶領域は最大で

$$(4 + 120 \times 2 + 4) \times 10^4 \approx 2.5MB$$

となる。

さらに、ネクスト動画リストは、全動画数を一万、各動画毎に記録できる次動画の最大数を10、動画名を120文字の2バイト文字、回数を4バイトに記録すると、必要となる記憶領域は最大で

$$120 \times 2 + (120 \times 2 + 4) \times 10 \times 10^4 \approx 27MB$$

となる。

よって、提案手法においてスーパーノードで必要となる記憶領域は $2.5 + 2.5 + 27 = 32MB$ となる。ここで、動画のビットレートが320kbps、動画の長さが180秒の場合、一つの動画サイズは約7MBである。したがって、スーパーノードで必要となる記憶領域は、動画4.5個分の大きさに相当する。これは、現在の一般的なコンピュータならば、十分許容できると考える。

5. ま と め

本論文ではP2P型ストリーミング配信において、再生開始までの待ち時間の短縮手法と、効率の良い検索手法について検討し、シミュレーション実験で検証を行った。シミュレーション実験により、本手法の次動画の先読みバッファリングが有効であることが分かった。

今後は、提案手法の検索効率向上のためのキャッシュノードリスト機能とキャッシュ判断の動作をシミュレータ上に実装し、効果を検証する必要がある。さらに、動画データの分割・復元・再生機能を加え、実環境での動作検証を行いたい。

参 考 文 献

1) GONG, A., DING, G., DAI, Q. and LIN, C.: BulkTree: an overlay network architecture for live media streaming, *Journal of Zhejiang University SCIENCE A*, pp.125-

130 (2006).

- 2) Yong, L., Yang, G. and Chao, L.: A survey on peer-to-peer video streaming systems, *Peer-to-Peer Netw Appl*, pp.18-28 (2008).
- 3) Chow-Sing, L. and Wei-Ting, S.: A fine-grained balancing scheme for improved scalability in P2P streaming, *Multimed Tools Appl*, pp.71-90 (2010).
- 4) Chao, L., Yang, G. and Yong, L.: Investigating the Scheduling Sensitivity of P2P Video Streaming: An Experimental Study, *Multimedia, IEEE Transactions on*, Vol.11, pp.348-360 (2009).
- 5) 後藤佑介, 鈴木健太郎, 義久智樹, 金澤正憲: P2P ストリーミング放送における待ち時間短縮のためのピア選択手法, 電子情報通信学会, DEWS, 第19回 (2008).
- 6) 藤本貴也, 遠藤 伶, 松本 敬, 重野 寛: P2P VoD サービスにおける動画の人気度に基づくキャッシュ手法の提案, 情報処理学会研究報告, DPS, 143号, No.24 (2010).
- 7) 総務省: P2P ネットワークの在り方に関する作業部会報告書, ネットワークの中立性に関する懇談会, WG2 P2P ネットワークの在り方に関する作業部会, (2007).
- 8) 上野真弘, 大坐島 智, 川島幸之助: 高次数をもつ非協力ピアの参加が非構造化 P2P ファイル共有ネットワークへ及ぼす影響の評価, 電子情報通信学会, 信学技法, NS2008-48.
- 9) 銭飛: NS2 によるネットワークシミュレーション第1版, 森北出版株式会社, (2006).