

## P2P を用いた VOD サーバの負荷低減方式の提案

島田 佳広<sup>†</sup> 佐藤 陽一<sup>†</sup> 中西 亮<sup>†</sup>  
品川 高廣<sup>††</sup> 吉澤 康文<sup>††</sup>

† 東京農工大学大学院工学教育部 †† 東京農工大学大学院共生科学技術研究部

大量の動画像情報をディスクから読み込み、クライアントに送信する VOD サーバでは人気タイトルの情報をメモリ内にキャッシュする工夫によりディスク I/O の負荷を低減することができる。しかしこの工夫が成功しても、通信回線を通して送信するオーバーヘッドの低減はできない。そこで、P2P プロトコルを用いて、クライアントの通信回線、CPU、メモリ等を利用し、VOD サーバの負荷を低減する方式を提案する。本方式は、人気タイトルに要求が集中して発生する際に有効と考える。本稿では、本方式を採用した VOD サーバとクライアントの処理方式を報告する。

**キーワード** Video On Demand, ストリーミング配信, キャッシュ管理, Peer-to-Peer

## Load Reduction Method of VOD Servers using P2P

Yoshihiro Shimada<sup>†</sup> Yoichi Sato<sup>†</sup> Ryo Nakanishi<sup>†</sup>  
Takahiro Shinagawa<sup>††</sup> Yasufumi Yoshizawa<sup>††</sup>

† Graduate School of Technology, Tokyo University of Agriculture and Technology †† Institute of Symbiotic Science and Technology, Tokyo University of Agriculture and Technology

VOD servers that read a large amount of dynamic image data and send it to clients can reduce load of disk I/O with caching data of popular title. However, even if this attempt succeeds, overhead caused by transmitting data via communication lines cannot be reduced. Then, we propose the method to reduce the load of VOD servers by using the P2P protocol, and using client's network bandwidth, CPU power, and memory etc. Our method is effective, when the demand concentrates on a popular title. In this paper, We report how VOD server and the client will work with our method.

**Key words** Video On Demand, Media Streaming, Cache management, Peer-to-Peer

### 1. はじめに

Video On Demand (以下、VOD と記す) サーバでは大容量のファイルを多数のクライアントへ配信する必要があるため、ディスクとネットワークの高速な入出力機構が要求される。ディスクの入出力においては、ディスク上のファイルをメモリ上にキャッシュすることで、入出力回数の削減が期待できるが、ストリーミ

ングメディアのファイルは、1GB ~ 数 GB もの容量になるため、複数のストリーミングメディアをサーバのメモリにキャッシュすることは難しい。一方、ネットワークの入出力においては、マルチキャストを用いて複数のクライアントに同時配信することで入出力回数の削減が期待できるが、ルータなどのネットワーク機器がマルチキャストに対応している必要があるため、容易に導入することはできない。

従来のクライアント/サーバ（以下、C/S と記す）モデルによる VOD 配信システムは、すべてのクライアントに直接サーバがコンテンツを配信するため、ユーザ数の増加にともないサーバやサーバ近傍のネットワークにおけるトラフィックの増加、またサーバ自身に大きな負荷が生じていた。そこで、近年 P2P 通信技術を用いた新しい VOD 配信技術が提案され、多くの研究が行われている [1, 2, 3]。P2P 型通信では、ピアと呼ばれる個々のホストが互いに直接情報をやり取りすることができる。P2P を用いた VOD 配信 (P2P-VOD) では、クライアントが一度受信したストリーミングメディアを別のクライアントへ再配信することによって、サーバが直接クライアントに対してコンテンツを配信することなくサービスを提供できるため、サーバの負荷を低減することができる。

しかし P2P-VOD では、クライアントも配信システムの一部を担うため、クライアントの振る舞いがそのままシステムに影響を及ぼしてしまう。転送を行っているクライアントがダウン（離脱もしくは故障）した場合、そのクライアントから配信を受けていたクライアントにはデータが届かない。

本稿ではチャンクグループという概念を用いてクライアント間における配信をサーバが管理する手法について述べる。本手法において、クライアント間での配信の実現による配信システム全体での性能向上と配信クライアントのダウン時における再生保証の実現を目指す。

以下、本稿では 2 章で P2P-VOD に対する要求分析について述べ、3 章では P2P を利用したストリーミング配信システムについて述べる。4 章ではシミュレーションによる本提案の効果予測結果について述べる。5 章では関連研究について述べる。

## 2. 要求分析

本章では、P2P-VOD を設計するために考慮すべき特徴としてストリーミングメディアの特徴と、P2P-VOD への要求について述べる。

### 2.1 ストリーミングメディアの特徴

P2P-VOD の実現に関するストリーミングメディアの特徴として (1) シーケンシャルアクセス特性、(2) 参照頻度の不均一性、(3) リアルタイム性があげられる。

#### (1) シーケンシャルアクセス特性

クライアントはファイルの先頭から終端方向へシーケンシャルにアクセスするため将来のアクセスが予測可能である。さらに、同じコンテンツへアクセスするクライアント同士は、常に一定の間隔を保ったままアクセス位置を移動するため、クライアント間での転送による配信が可能となるときは、将来に渡って配信を受けられる可能性が高い。

#### (2) 参照頻度の不均一性

人気コンテンツに対する再生要求が圧倒的に多く、コンテンツに対する参照頻度が一様ではない。参照頻度の高いコンテンツの配信をクライアント間で実現することができればサーバの負荷の大幅な低減が期待できる。

#### (3) リアルタイム性

遅延やジッタがクライアントでの再生品質に影響を及ぼすため、リアルタイム性が重要である。リアルタイム処理には処理時間の見積もりが重要であるが、たとえばクライアント-サーバ間のネットワークにおける RTT (Round Trip Time) をあらかじめ計測しておくことで配信ピアの障害時に柔軟な対応が可能になる。

## 2.2 P2P-VOD への要求

P2P-VOD が満たすべき要件として次のような点があげられる。

#### (1) 対故障性

ユーザの使用する PC などを使用してシステムを構築するため、ピアのダウンに対応できなければならない。他のピアに転送しているピアがダウンすると、ダウンしたピアから直接配信を受けているピアだけでなく、下位のすべてのピアが配信を受けられなくなる。そのためストリーミング映像を途切れさせないためのリアルタイム性のあるフェールセーフ機構が必要である。たとえば、常にピアの持つ未再生データ量を把握し、ある一定量を下回ったらサーバへ再接続要求を送信する方法がある。この際、未再生データ量をどの程度に見積もるのが問題となる。

#### (2) 映像品質の保証

ストリーミングメディアの送信は通常 UDP によって行われるため、パケットが落ちた場合、映像品質が落ちてしまう。さらに、1 度落ちたパケットは下位のピアに転送されないため、映像品質の低下が下位ピアに連鎖してしまう。そこで映像品質を保証するストリーミングメディアの転送方式が必要となる。

#### (3) キャッシュサイズの異なるクライアントの対応

クライアントがキャッシュとして利用可能なメモリサイズはそれぞれ異なるため、サーバは各クライアントが利用可能なメモリサイズを把握して、可能な限りクライアント間での配信を増やしサーバの負荷を低減したい。

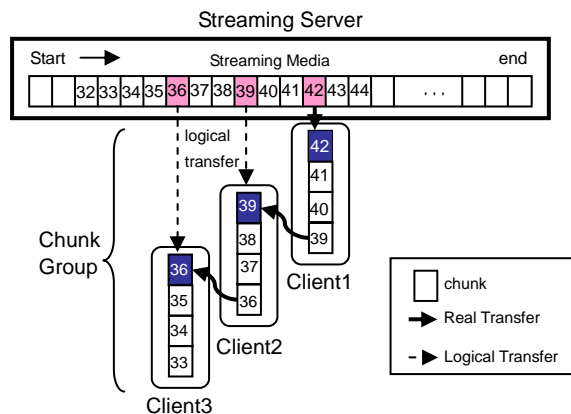


図 1 VOD 配信システム

### 3. P2P を利用したストリーミング配信

本章では P2P を利用した VOD ストリーミングについて説明する。

#### 3.1 システム概要

想定する VOD 配信システムの全体像を図 1 に示す。クライアント 1 のみサーバから直接配信を受け、クライアント 2,3 はそれぞれ先行するクライアントからコンテンツの配信を受けている。

ストリーミングメディアは MPEG2-PS 形式を想定し、チャンクという単独で再生可能なデータ単位でコンテンツの転送を行う。チャンクはコンテンツの先頭からチャンクサイズごとにシーケンシャルに番号が割り振られている(Chunk Number)。コンテンツの参照位置が近く、キャッシュの転送を行うクライアントのグループをチャンクグループと呼ぶ。図 1 ではクライアント 1,2,3 がチャンクグループを形成している。

クライアントはコンテンツを外部記憶に保存せずに、チャンク単位でメモリ上に保持しなければならない。サーバは使用可能なメモリサイズに応じて、クライアントのメモリに同時にキャッシュするチャンク数(MCC: Max Number of Chunks on Client Memory)を定める。MCC は障害時の再生保証を考え、3 以上の自然数をとる。クライアントのキャッシュ容量に関して以下の式が成り立つ。

$$\begin{aligned} \text{Available Memory Size of Client} \\ = \text{Max Chunk size} \times N \quad (1) \\ (N \geq 3, N \text{ は自然数}) \end{aligned}$$

またクライアント - サーバ間、クライアント - クライアント間では、セッションごとに、次に述べる複数のコネクションを利用して通信を行う。クライアントは SCP(Session Control Protocol)を利用して、ストリームの再生、停止などの操作をサーバへ要求し、ストリーミングメディアデータは DTP(Data Transfer Protocol)を利用して配信される。またサーバは SCPを利用して、ストリーミングメディアデータの転送を

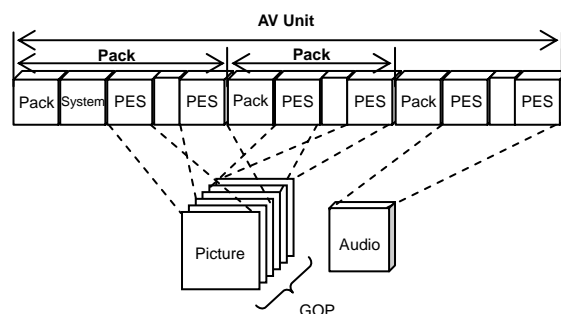


図 2 MPEG2-PS システム内部構造

クライアントへ要求し、クライアントは指定された他クライアントへ DTP を利用してデータの転送を行う。

#### 3.2 チャンク

本節ではチャンクの内部構造と、チャンクサイズについて述べる。

##### 3.2.1 AV Unit

図 2 は MPEG 2-PS システムの内部構造である。MPEG ビデオは動き補償によるフレーム間予測の技術を用いて、前後の画面を元にして作られるので、複数のピクチャをひとまとまりにした GOP(Group of Picture)構造を再生の単位として用いている。

また MPEG ではビデオとオーディオを同期し、一本化するために MPEG システムを提供している。ビデオデータやオーディオデータは PES(Program Elementary Stream)パケット単位で区切れ、さらに複数のパケットを束ねたパックと呼ばれる構成単位でデータを取り扱う。

本稿ではオーディオデータを含む 1GOP 分のパックをまとめて、AV Unit と呼ぶ。AV Unit は単独で再生可能な単位であるので、チャンクは複数の AV Unit で構成される。

##### 3.2.2 チャンクサイズ

前項で述べたとおりチャンクは複数の AV Unit からなるため、チャンクサイズにおいて以下の式が成り立つ。

$$\begin{aligned} \text{Chunk Size} = \text{AV Unit Size} \times N \quad (2) \\ (N \text{ は自然数}) \end{aligned}$$

MPEG ではフレームレートやビットレートは定められているが、GOP 単位の再生時間に明確な基準が無いため、GOP ごとに再生時間にばらつきがある。提案方式では 1 チャンクあたりの再生時間(チャンクタイム)を基準にシステムを設計しているため、コンテンツごとにチャンクタイムを一定の範囲に収めることが望ましい。そこで想定する MPEG2-PS システムの AV Unit を、チャンクタイム(T(chunk)={1,2,3,4,5})を満たすようにコンテンツの先頭から結合したときのチャンク

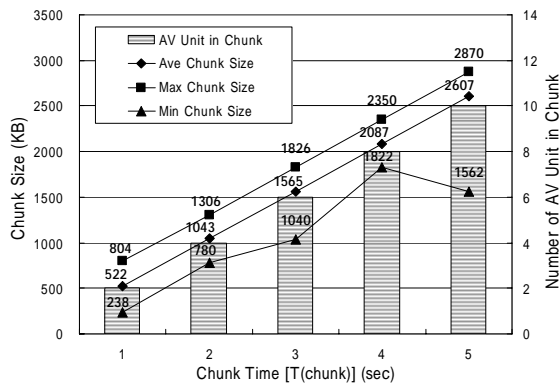


図 3 チャンクサイズ

表 1 ブロックの状態

状態	意味
Init	チャンク未受信
Past	再生完了
Current	再生中
Future	未再生

表 2 ブロックの転送状態

状態	意味
Forwarded	転送完了
Not-forwarded	未転送

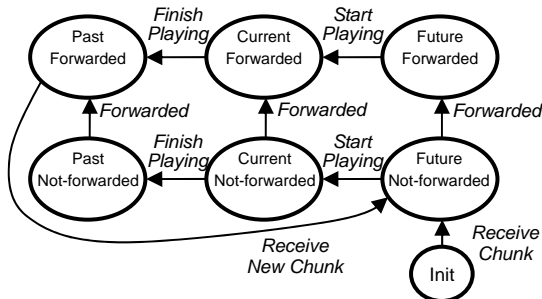


図 4 ブロックの状態遷移

クサイズの平均値、最大値、最小値と1チャンクに含まれる AV Unit 数の平均値を図 3 に示す。なお固定ビットレート(4Mbps)のストリーミングメディアを使用し、その時の AV Unit の平均サイズは 261KB である。図 3 から最大チャンクサイズ(MAX Chunk Size)がほぼ一定の範囲内に収まっていることがわかる。クライアントは最大チャンクサイズを基準にして、使用するメモリ容量を決定する。以下にその例を示す。

ex. MCC=3, T(chunk)=5 のとき

(式 1), (図 3)より

$$\begin{aligned}
 \text{Available Client Memory Size} &= \text{Max Chunk size} \times N \\
 &= 2870 \times 3 = 8610 \text{ (KB)}
 \end{aligned}$$

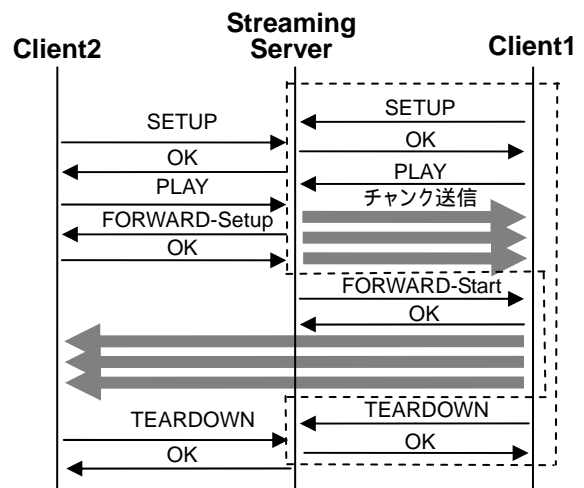


図 5 プロトコルシーケンス

クライアントはチャンクタイム 5 秒のコンテンツを視聴するとき、最低でも 8.4MB のメモリ領域を確保しなければならない。これらの数字は個人使用の PC のスペックや専用機器を考えたとき十分に適用可能な数値であるといえる。

### 3.3 クライアントにおけるキャッシング

本システムにおけるクライアントは、チャンクを他クライアントに転送する必要があるため、再生の終了したチャンクをすぐに破棄することはできない。また、クライアントはメモリにのみチャンクをキャッシュし外部記憶を使用しない。このため、クライアントは容量の限られたメモリにおいて、不要となったチャンクを適切に廃棄し、新たなチャンクを受信するためのメモリを確保する必要がある。

クライアントはブロックと呼ばれる単位でキャッシングを管理する。1ブロックには1チャンクが格納される。ブロックは表 1 に示す 4 つの状態 (Init, Future, Current, Past) と表 2 に示す 2 つの転送状態 (Forwarded, Not-forwarded) を持ち、図 4 に示す状態遷移を行う。

Init ブロックはキャッシュとして有効なデータを持たないブロックである。チャンクを受信しブロックに格納すると Future ブロックとなる。Future ブロックは再生が開始されていないことを示す。再生を開始すると Future ブロックを Current ブロックとする。Current ブロックは現在再生しているブロックを示す。よって Current ブロックは常に 1 つである。再生が終了すると Past ブロックとなる。また、3 つの再生状態 (Future, Current, Past) のそれぞれにおいて転送状態を持つ。Not-forwarded は転送していないことを示し、転送が完了したブロックは Forwarded となる。Past かつ Forwarded のブロックは、不要であるため、新たなチャンク受信するためのブロックとして再利用される。

### 3.4 制御プロトコル

提案システムを実現する制御プロトコルはセッションを管理する SCP とデータ転送を行う DTP で構成される。SCP は TCP 上で DTP は UDP 上で実現される。

#### 3.4.1 SCP (Session Control Protocol)

SCP は RTSP(Real Time Streaming Protocol) に似たプロトコルであり、4 つのメソッド (SETUP, PLAY, FORWARD, TEARDOWN) を持つ。図 5 に SCP のプロトコルシーケンスを示す。

まずクライアントによる転送を行わない場合の配信について述べる (図 5: 点線部分)。SETUP メソッドは配信コンテンツ名、ポート番号、クライアントメモリサイズ等、本システムにおけるコンテンツ配信に関する基本的な情報をやり取りするために使用される。

OK でサーバはチャンクサイズ、クライアントメモリに保持するチャンク数、チャンクの総数等を送信し、そのパラメータに応じてクライアントは受信準備を行う。クライアントは受信準備が完了したら、サーバへ

PLAY メソッドを送信し、コンテンツの配信を待ち、コンテンツを再生する。コンテンツの再生終了後、クライアントはサーバへ対して TEARDOWN メソッドを送信し、その応答 (OK) を受信してシステムから離脱する。

次にチャンクグループを形成し、チャンクを下位ピアに転送する配信について述べる。ここでは、先にコンテンツ配信を受けているクライアントをクライアント 1、後から同一コンテンツへの配信要求を行うクライアントをクライアント 2 とする。PLAY メソッド送信後クライアントは同一コンテンツを視聴している直前のクライアントの状況に応じて 3 つの状態 (OPEN, SETUP, WAIT) を持つ。各々については後節で詳しく説明するが、この状態が OPEN だったときに限り、下位ピアへの転送要求を受け付けることができる。

サーバは OPEN 状態のクライアント (クライアント 1) がいるコンテンツへの PLAY (受信準備完了) を受け付けると、PLAY を送信したクライアント (クライアント 2) へ対して、FORWARD-Setup メソッドを送信する。FORWARD-Setup メソッドは上位ピアの IP アドレスや Chunk ACK パケット、Chunk WIN パケットの宛先ポート番号等のチャンク転送に必要な情報をやり取りする。転送受付準備が完了したら、クライアント 2 はサーバに対して OK を送信し、サーバは次にクライアント 1 に対して転送開始命令である FORWARD-Start メソッドを送信する。FORWARD-Start メソッドを受信したクライアント 1 はクライアント 2 に対してチャンクの転送を行う。

#### 3.4.2 DTP (Data Transfer Protocol)

DTP はチャンク単位でのデータ転送を行うためのプロトコルで、コンテンツデータ送受信部と Chunk ACK (チャンク受信完了応答)、Chunk WIN (チャンク単位のウィンドウサイズ) パケットを送信する制御部で構成される。

表 3 クライアントの状態

状態	意味
CLOSE	転送要求を受け付けていない
OPEN	転送要求を受け付けている
SETUP	チャンクグループ作成処理中
WAIT	待ち状態

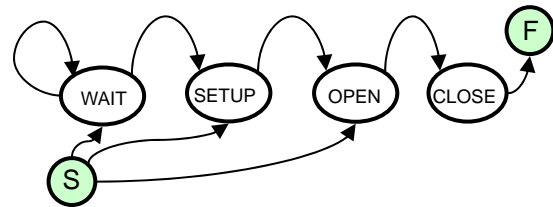


図 6 クライアントの状態遷移図

データ送信部は下位クライアントからの Chunk WIN を受信して、次のチャンクを送信する。Chunk WIN は前節で述べた不要なブロックが生じたときに送信する。またクライアントはチャンクの受信が完了したら、送信元に対して Chunk ACK を送信する。これらの Chunk ACK、Chunk WIN を利用して、下位ピアのキャッシュの空き容量を考慮した効率的なデータ転送を行うことができる。

### 3.5 チャンクグループ

#### 3.5.1 チャンクグループ作成アルゴリズム

チャンクグループ作成においてクライアントは表 3 に示す 4 つの状態 (CLOSE, OPEN, SETUP, WAIT) を持ち、図 6 に示す状態遷移を行う。CLOSE はチャンク転送要求を受け付けていない状態である。OPEN 状態のクライアントはある一定時間転送要求を待ち、転送要求が来れば下位ピアに対して転送を行い、転送要求が来なければ CLOSE 状態になる。この下位ピアへの転送要求を待つ時間の最大値を MTAf (Max Time to Accept Forward Request) と呼ぶ。MTAF の決定法については次節で説明する。SETUP 状態は転送準備中を示す。SETUP 状態のクライアントは転送要求が受け入れられ、上位ピアからチャンクの配信を受けるか、転送要求が失敗すると OPEN 状態になる。WAIT 状態は直前のクライアントが SETUP 状態か WAIT 状態の時に限り、待ち行列を作って待つ。直前のクライアントが OPEN 状態になったときに SETUP 状態になる。WAIT 状態は通常の配信ではほとんど見られないが、Flash Crowd 等の急激なリクエストの増加時には必要になる。

図 7 に同一コンテンツを視聴するクライアント (C1, C2, C3) のリクエストが到着したときの、チャンクグループ作成の様子とチャンク転送について示す。C1 が OPEN 状態の時に C2 の受信準備が終了したため、C2 は SETUP 状態になる。C2 の SETUP 中に C3 の受信準備が完了したため WAIT 状態になる、その後は図 5 に示した状態遷移を行っていく。

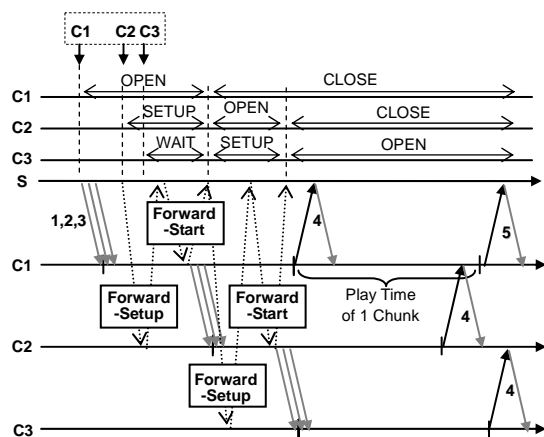


図 7 チャンクグループ作成の全体像

### 3.5.2 MTAF

前節で説明したようにチャンクグループはセッション開始時でのみ作成される。そこでクライアントは OPEN 状態において、後続クライアントへの転送のために再生の終了したチャンク(Past Chunks)をキャッシュしておく必要がある。チャンクをいくつか受信し、ブロックが飽和した状態では、先頭チャンクをキャッシュしているかぎり空きブロックが生まれなため、新たなチャンクを受信できなくなる。そこで、先頭チャンクのキャッシュに保存期限を設ける。この時間が MTAF であり、MTAF を過ぎるとクライアントは転送要求を受け付けない。

転送要求を受け付ける時間を最大限に長くするため、未再生のデータ量がクライアントの再生を保証する最低限に減少するまで先頭チャンクをキャッシュしておく、再生開始からのこの時間を MTAF とする。MTAF 以内に転送要求を受信したとき、クライアントは残りのデータを再生する間に下位ピアに対する転送と今後再生するための新たなチャンクを受信を行わなければならない。今回 MTAF は未再生のチャンクが 1 つになるまでの時間とした。

## 3.6 再生保証機能

本節では、ストリーミングメディアの送信におけるパケット落ちや、転送を行うピアのダウンに対応するための再生保証機能について述べる。

### 3.6.1 サーバへの再接続

ストリーミングメディアの転送を行うピアは、サービスの途中でダウンする可能性がある。そこで配信ピアのダウン時においてもサービスを継続するため、ピアから配信を受けるピアは、配信ピアのダウンを検出してサーバへ再接続する。ピアは自分のキャッシュに持つ未再生データの再生時間が、あるデッドラインを下回った場合、配信ピアがダウンしたとみなす。この

ときのデッドラインはサーバへの再接続とサーバからチャンクを受信するために必要な最低限の時間である。

デッドラインは、(1) クライアントが再接続要求を送信する時間、(2) サーバが再接続要求を処理する時間、(3) サーバのディスク入出力時間、(4) チャンクの送信時間の合計で求められる。このときサーバは配信ピアのダウンによる再接続要求を優先して処理する。

### 3.6.2 パケット再送機構

ストリーミングメディアの送信は UDP により行われるため、パケットが落ちる場合がある。パケットが落ちると MPEG の性質上、映像にノイズが入ってしまう。これを避けるためパケット落ちを検知してパケットの再送を試みる。クライアントはパケット落ちを検知した場合、配信元のピアもしくはサーバに対して NACK を送信する。配信元は NACK を受信したとき、該当パケットを再送する。繰り返し NACK を送信しても再送が行われずに配信ピアダウンのデッドラインを迎えた場合、配信ピアはダウンしたとみなす。

## 4. シミュレーションによる効果予測

提案方式の有効性を検証するために、ストリーミングサーバに対する負荷モデルを定義し、シミュレータによる効果予測を行った。本章ではシミュレーション環境と負荷モデルの定義、そして得られた結果と考察について述べる。

### 4.1 負荷モデル

シミュレータはポアソン分布に従ったリクエスト到着間隔でセッション開始のリクエストを発生させる。各セッションにおいて再生するコンテンツは負荷モデルごとの参照頻度に従い選択されるが、その前後関係に依存性は無いものとする。また各セッションはコンテンツの最初から最後まで通常再生するものとし、途中で再生を停止したり、早送り、巻戻しなどの特殊再生を行わないものとする。

コンテンツの参照頻度モデルとして Zipf 近似分布に従う Zipf モデルを使用する。Zipf 近似分布では、コンテンツを参照頻度順にインデックス付けた場合、 $i$  番目のコンテンツを参照確率が  $1/i^{(1-\alpha)}$  ( $0 < \alpha < 1$ ) となる。これは人気順位とアクセス数の積が一定になることを意味し、 $\alpha$  が 0 に近づくほど参照が偏ったスキューな分布になり、1 に近づくほど一様分布に近くなる。Dan ら[5]はレンタルビデオの貸し出し数は  $\alpha = 0.271$  の Zipf 近似分布になると報告している。レンタルビデオにおける貸し出し傾向はストリーミングサーバに類似すると考えられるため、Zipf モデルでも  $\alpha = 0.271$  の Zipf 近似分布に従うものとする。

ストリーミングメディアは表 4 に示すような MPEG2-PS ファイルのコンテンツを 100 本とする。そして図 8 に想定する環境下 ( $\alpha = 0.271$ , コンテンツ数 = 100) における参照累積分布を示す。円グラフの各領域は参照頻度が高い順に 10 等分されており、Zipf モ

表4 ストリーミングメディア

フォーマット	MPEG2-PS
再生時間	120分
画像サイズ	720×480
ビットレート	4Mbps(固定)
ファイルサイズ	3.6GB

表5 シミュレーションパラメータ

T(cset)	転送準備応答待ち時間
T(pset)	転送開始応答待ち時間
T(chunk)	平均チャンクタイム
MCC	クライアントのキャッシュ容量
MTAF	チャンク転送要求受付時間
C	チャンクサイズ
$\tau$	リクエスト到着間隔の平均
	Zipf 近似分布のパラメータ

デルでは上位 10%のコンテンツへの参照が 47%を占めていることが分かる。

## 4.2 パラメータ

VOD 配信サーバのシミュレーションに対するパラメータを表5のように定義した。

T(cset)は、クライアントにチャンク転送準備要求を出してから応答が返ってくるまでの時間、T(pset)はクライアントにチャンク転送要求を送信してから応答が返ってくるまでの時間である。T(cset),T(pset)共に 10msec とした。T(chunk)は、チャンクタイム(1チャンクあたりの再生時間)の平均である。クライアントは1つのチャンクの受信が完了すれば、T(chunk)分のストリーミングコンテンツの再生が可能になる。MTAFは前節で述べた、チャンクグループの最下位のピアが子ピアへの転送要求を受け付ける時間の最大値である。MTAFはT(chunk)に伴って変化する。

そして平均リクエスト到着間隔  $\tau = 3$  秒の場合を高負荷時、 $\tau = 9$  秒の場合を低負荷時と定義する。

$\tau = 3$  の条件は提案方式等を評価するための極端な数値ではあるが、Flash Crowd 状態ではリクエストがこのような到着間隔で来ることもある。

## 4.3 結果と考察

提案方式の有効性とチャンクサイズ(C)による影響に関して、シミュレーションによる効果予測を行った。

### 4.3.1 チャンクグループ作成による負荷低減

提案方式の有効性を示すために高負荷時と低負荷時における配信クライアントの総数とチャンクグループを形成しているクライアント数の関係を図9、図10に示す。ここでは参照頻度の上位10%のクライアント数の違いを分かり易くするために、Y軸を対数メモリにしている。また、チャンクグループを形成しているクライアントの割合の測定を行うために、あえてI/Oやネットワーク環境のボトルネックは考えないものと

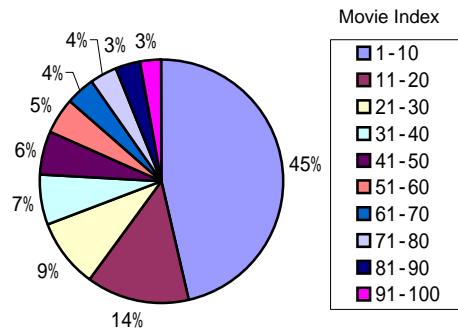


図8 参照累積分布

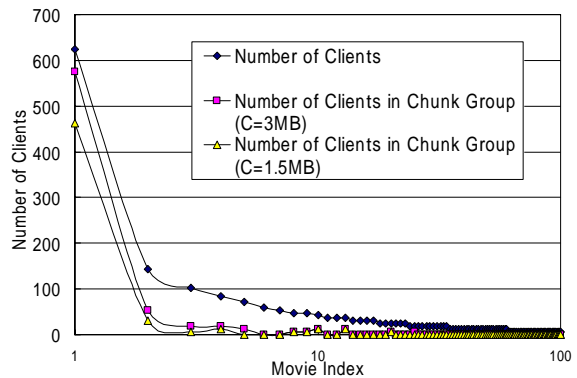


図9 配信クライアント数とチャンクグループ (高負荷時)

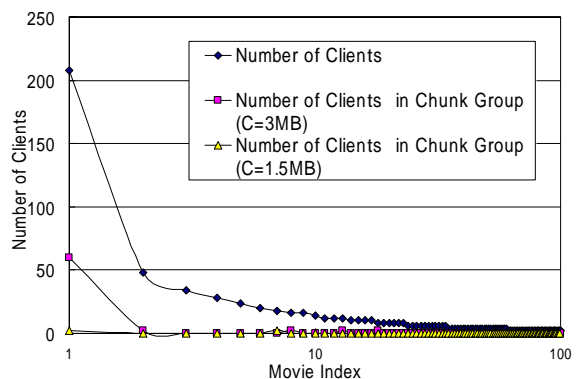


図10 配信クライアント数とチャンクグループ (低負荷時)

する。

Movie Index 別に見ると、高負荷時で最大 93%、平均 32%(共に C=3MB の時)、低負荷時で最大 29%、平均 9%(共に C=3MB)のクライアントがチャンクグループを形成していることがわかる。チャンクグループを形成しているクライアントはサーバから直接配信を受けずに上位のクライアントから配信されるため、その分サーバは直接配信を行うクライアント数を増やすことができる。

表6 最大配信クライアント数

	高負荷時	低負荷時
C=3MB	375	277
C=1.5MB	316	261

#### 4.3.2 最大配信クライアント数

想定する環境下で提案方式実効した場合の最大配信クライアント数の増加について述べる。ディスクの実効バンド幅を 125MB/sec、ネットワークバンド幅を 1Gbps と想定すると、理論上 4Mbps の MPEG2 コンテンツは最大で 250 ストリーム同時にユニキャスト通信により配信することができる。

表6はサーバから直接コンテンツを配信しているクライアント数が 250 に達したとき、システム全体でコンテンツの配信を受けているクライアント数をシミュレーションしたものである。

C=3MB の場合、高負荷時で 1.5 倍、低負荷時で 1.11 倍、最大配信クライアント数が増加した。また C=1.5MB の場合も、高負荷時で 1.26 倍、低負荷時で 1.04 倍増加した。この結果は提案方式の有効性を示唆していると言える。

特に高負荷時に効果を発揮していることから、近年に問題になっている Flash Crowd[6]対策の VOD サーバとしても用いることができる。

## 5. 関連研究

近年では、P2P 環境において、ピア間でストリーミング配信を中継することで実現する VOD システムに関する多くの研究がなされている[1, 2, 3]。本提案手法は、サーバがクライアントを集中管理することで実現しているが、P2P 環境における VOD システムとも技術的な共通点が多い。P2VoD[1]は、同一コンテンツの配信を受けるピアのうち、時間的に距離の近いピアを Generation と呼ばれるグループにする。こうして転送先ピアへの配信経路を冗長化することでサーバを介さない障害復帰を可能としている。本提案手法では、障害時にクライアントはサーバへの接続を必要とするが、経路の冗長化が必要ないためサーバの配信数は少なくなる。

Hefeeda らは、ピアのストレージを使用するストリーミング配信システムを提案している[2]。ピアはセグメントに分割されたストリーミングメディアをストレージにキャッシュし、自分の持つセグメントの配信を受け持つ。ピアのメモリに一時的にキャッシュされたストリーミングメディアを使用するのではなく、ストレージを使用してメモリよりも多くのストリーミングメディアをキャッシュすることで、サーバの負荷の軽減を図る点が本提案手法と異なる。しかし、ピアは本来の配信サービスには不要なリソースを配信システムに提供するため、ピアに対する動機付けが必要である。

また近年では、プロキシを用いたインターネットにおけるグローバルな負荷分散やストリーミングメディ

アのキャッシュ分散手法などが研究されている[4]。これらの研究においても、P2P 環境におけるストリーミング配信とは共通点が多い。

## 6. おわりに

本稿では、ストリーミングメディアの参照特性とクライアントのキャッシュに注目したチャンクグループによるキャッシュオフロード方式を提案し、シミュレーションによる有効性の検証を行った。本システムは、クライアント間における配信により、サーバの負荷を低減し、さらに転送を行うクライアントがダウンした場合においても、サーバへの再接続により映像品質を保証する機能を持つ。シミュレーション結果より、本方式はクライアントキャッシュ容量が 12MB のとき、高負荷時で 1.5 倍、低負荷時で 1.1 倍の配信クライアント数の増加が期待できるという予測を得た。

なお現在本機構を実装中であり、今後の課題として実環境下での評価、および既存研究との比較評価があげられる。

## 参考文献

- [1] T. Do, K. Hua, and M. Tantaoui. P2VoD: Providing Fault Tolerant Video-on-Demand Streaming in Peer-to-Peer Environment. Proc. of the IEEE Int. Conf. on Communications (ICC 2004) 2004.
- [2] M. Hefeeda, and B. Bhargava. On-Demand Media Streaming Over the Internet. In Proc. of the 9th IEEE Workshop on Future Trends of Distributed Computing Systems (FTDC03) 2003.
- [3] S. Shimizu, and T. Nakamura. Reliable Multicast based on Peer-to-Group Dissemination. IBM Research Report RT0475. 2002.
- [4] B. Wang, S. Sen, M. Adler, and D. Towsley. Optimal Proxy Cache Allocation for Efficient Streaming Media Distribution. IEEE Transaction on Multimedia, Special Issue on Streaming Media, Vol. 6, Issue 2, pp. 366-274, April, 2004.
- [5] A. Dan, and D. Sitaram. Buffer Management Policy for an On-Demand Video Server. IBM Research Report RC 19347,1994.
- [6] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long. Modeling Analysis and Simulation of Flash Crowds on the Internet. Technical Report UCSC-CRL-03-15. 2004.
- [7] 高野了成, 浅見和男, 帆波幸二, 吉澤康文. ストリーミングサーバの参照特性に基づく入出力削減方式. 情報処理学会論文誌 Vol.45 No.4 pp.1090-1101. 2004.