

スケジュール化された Push 型配信を可能とする VoD システムの検討

知花 昌樹[†] 応和 大輔[†] 高木 真一^{††} 富永 英義^{†,††}

[†] 早稲田大学大学院国際情報通信研究科
〒 367-0035 埼玉県本庄市西富田大久保山 1011
^{††} 早稲田大学国際情報通信研究センター
〒 367-0035 埼玉県本庄市西富田大久保山 1011

E-mail: †{chibana,owa,takagi}@tom.comm.waseda.ac.jp, ††tominaga@waseda.jp

あらまし 本研究は、従来のユニキャストによる VoD 配信アーキテクチャに、Push 型配信の機構を導入し、大規模ユーザ収容を実現するシステムを提案する。従来のユニキャストによる VoD 配信では、大規模ユーザ数を収容すると、オンデマンドアクセスの集中によるネットワーク負荷が発生し、受信者数制限などのサービス低下が避けられないのが現状である。提案システムでは、クライアントの映像コンテンツダウンロードタイミングを分散することで、ネットワークトラフィックを平滑化し、かつ、配信可能ユーザ数の変動にもスケラブルに対応することができる。

キーワード Push 型配信, 映像コンテンツ配信, ビデオオンデマンド, 公開日前配信

A Study on VoD Systems with Scheduled Push Delivery

Masaki CHIBANA[†], Daisuke OWA[†], Shin'ichi TAKAGI^{††}, and Hideyoshi TOMINAGA^{†,††}

[†] Graduate School of Global Information and Telecommunication Studies, Waseda University
1011 Okuboyama Nishi-Tomida Honjo-shi Saitama 367-0035 Japan
^{††} Global Information and Telecommunication Institute, Waseda University
1011 Okuboyama Nishi-Tomida Honjo-shi Saitama 367-0035 Japan

E-mail: †{chibana,owa,takagi}@tom.comm.waseda.ac.jp, ††tominaga@waseda.jp

Abstract Recently, with the spread of broadband network, the contents delivery service by Internet has activated. The systems which proposes with STB of the HDD attachment receives and accumulates the contents which are delivered with Internet, connects STB to the television and views. As for differing from usual VOD, the user with on demand it downloads not only, it is the point which can do push delivery. The case of push delivery, reception of the contents list becomes the trigger, the contents are downloaded automatically. In this research, delivery systems of the contents list because the systems which proposes is actualized is proposed.

Key words Push Delivery, Video Contents Delivery, Vieo on Demand, Advance Release

1. はじめに

ブロードバンドネットワークの普及により、音楽や映像などの大容量リッチコンテンツを扱うことが可能となった。特に近年、ハイビジョン対応の大型ディスプレイなどの普及によって、専用受信端末 STB (Set Top Box) を用いたテレビ向けの映像配信の需要が高まりつつある。

本稿では、インターネットを使った映像配信システムにおける配信サーバ側のネットワークトラフィック輻輳発生を回避するため、クライアントの映像コンテンツダウンロードのタイミングを配信サーバ側でスケジュールできるシステムを提案する。本方式を利用することで、配信トラフィックの平滑化と一斉同時

公開を実現できる。

第 2 節で現状の課題と要件、関連研究について整理し、本研究のターゲットを明らかにする。第 3 節で提案 VoD システムの概要について説明し、システム要件を実現するための機能を記す。第 4 節では提案 VoD システムの基本設計について動作例も交えて説明する。第 5 節で実装システムについて詳述する。

2. 現状の課題と要件

2.1 現状の課題

IP ネットワークによる映像配信では、1 対 1 通信を行うユニキャストによる配信が主流となっている。ユニキャストによる配信では、ユーザのアクセスが増加するにつれ、配信サーバの

負荷が増加し、受信者数制限などのサービス低下に繋がる。そのため、大規模なユーザ数への配信を行う場合には、サーバ通信回線増強のため膨大な設備投資が必要となる。

多数のユーザに効率よく配信を行う手法として1対多通信であるマルチキャストによる配信がある。しかし、IPマルチキャストによる放送サービスは、通信経路上の全てのルータが対応している必要があり、特定通信事業者が管理するCDN(Content Delivery Network)と呼ばれる閉じたネットワーク内でのみサービスを行っている。そのため、ユーザにとってサービス加入への敷居が高く、大規模な普及には至っていないのが現状である。

2.2 ターゲット

我々の提案する映像コンテンツ配信システムは、ユーザ端末に映像コンテンツを蓄積するためのHDDを内蔵したSTBを利用する。配信ネットワークにはインターネットを利用し、映像コンテンツの配信形態は、コンテンツリストを用いたPush型のオンデマンドダウンロードである。

配信前に完成されたビデオデータが配信サーバに用意されていることを前提とするため、リアルタイム性を考慮しない映像コンテンツを対象とする。尺長としては数十分から百数十分の比較的長時間の尺長に対応する。映像フォーマットはH.264形式とし、ハイビジョン映像にも対応させる。

また、本研究で定義するPush型コンテンツ配信とは、クライアントがユーザの操作に因らず、バックグラウンドで自動的かつ定期的にコンテンツのダウンロードを行うものである。Push型コンテンツ配信の身近な例として、電子メールの自動受信(定期受信)やアプリケーションプログラムの自動アップデートなどがある。一般的なVoDシステムでは、ユーザのオンデマンドアクセスに対するレスポンスとしてビデオストリームを配信する。本研究では、映像コンテンツ配信にPush型配信の機構を導入し、サーバ主導で配信スケジュールを作成することを実現する。

2.3 要件

本研究では、以下の3つの要件を実現する映像配信システムを提案する。

配信サーバ負荷の平滑化

従来のVoDシステムでは、配信サーバアクセスが集中した際の最大負荷を考慮したネットワーク設備を導入する必要がある。配信サーバ負荷が導入設備の許容を超えるとサービスを享受できないユーザが発生するためである。しかし、配信サーバ負荷が最大となるのはごく限られた時間帯であって、通常時は最大時の2分の1程度のトラフィック量だと言われている[1]。本研究では、配信トラフィックを平滑化することによって配信ネットワークコストを抑えることを実現する。

大規模一斉公開の実現

映画の観客動員数は公開週の週末に集中し、その後は減少する傾向にあることから、VoDサービスにおいても公開直後に最もアクセスが集中することが予想される。本研究では、公開時のアクセス集中を避けるため、公開前に受信端末が自動で映像コンテンツのダウンロードを完了し、公開直後でも確実に視聴可能な環境を実現する。

ロングテール映像コンテンツ配信

低需要商品の販売促進に焦点を当てた新しいビジネスモデ

ルとして、ロングテールビジネスモデルが注目を集めている。iTunes Store [2] や Amazon [3] ではユーザが求める最適なコンテンツに誘導するWEBシステムを導入することでロングテール化を図っている。本研究では、映像コンテンツ配信においてユーザの趣味嗜好を考慮したオススメコンテンツの自動配信を実現する。

2.4 関連研究

映像配信サーバアクセス負荷を分散する手法として、キャッシュサーバを用いたCDNの構築がある[4]。この研究は、複数のキャッシュサーバをIX(Internet eXchange)や、ISP(Internet Service Provider)に設置し、ユーザ近傍のキャッシュサーバからビデオストリームの配信を行う。サーバ負荷分散と同時に、通信距離の短縮によるリアルタイム品質の向上を実現している。また、P2Pネットワークを利用し、各ノードに配信負荷を分散させる技術としてALM(Application Level Multicast)がある[5]。IPマルチキャストが対応ネットワーク上でのみ実現されるのに対し、ALMはアプリケーション層でマルチキャストを実現しているため特殊なネットワーク条件が不要となる。

本研究の要件と上記研究の一部は重複している。しかし、本研究ではCDNを必要としないインターネット配信であること、ストリーミングではなくダウンロード配信であること、マルチキャスト放送では対応できないロングテール映像コンテンツ配信を可能とする点で異なっている。

また、テレビ放送番組の付加情報を扱う番組連動型情報サービスにおいて、ユニキャストを用いた際のサーバアクセス集中を回避する研究がある[6]。この研究では、リアルタイム性の高い情報はPush型配信を用い、リアルタイム性の低い情報はPull型配信を用いている。さらにダウンロードタイミングの時間的分散を行うことでサーバアクセスの集中を回避している。

本研究の設計指針と上記研究の一部は類似しているが、本研究では番組連動情報よりも情報量の大きい映像コンテンツそのものを配信することに違いがある。

3. 提案 VoD システムの概要

提案VoDシステムでは、前述のシステム要件を実現するために主に3つの機能を導入する。

3.1 映像コンテンツ配信タイミングのスケジュール化

提案VoDシステムでは、ユーザ毎、映像コンテンツ毎にPush型配信のタイミングが配信サーバによってスケジュール可能である。これにより、ユーザのダウンロードアクセスを計画的に分散させることができる。すなわち、配信サーバ負荷の平滑化が可能となる。

3.2 公開日前配信

映像コンテンツに公開日の概念を設け、上記のスケジュール化において公開日前にダウンロードを完了するようにする。また、ダウンロード完了後でも公開日前に再生できないようにし、公開と同時に再生可能な状態とすることで、Push配信した全てのユーザへの一斉公開を実現する。従来のVoDサービスでは公開直後にサーバアクセスが集中してしまいが、提案VoDシステムではサーバアクセスの時間的分散が可能となる(図1)。

3.3 レコメンド配信

上記の公開日前配信において、配信サーバに存在する全

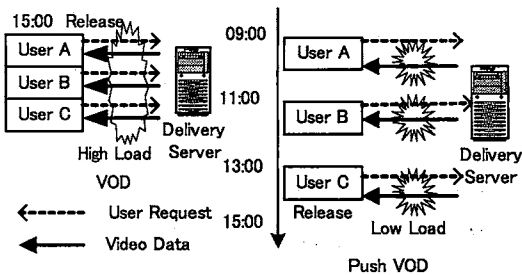


図1 VoD サービスと提案システムの配信サーバ負荷

ての映像コンテンツについて Push 配信してしまうと、ユーザが視聴しないコンテンツまで配信することになるので、従来のVoD サービスと比較して配信トラフィックが増加してしまう恐れがある。そこで提案 VoD システムは、ユーザの予測視聴確率が高い映像コンテンツに関しては公開日前配信を行い、ユーザの予測視聴確率が低い映像コンテンツは従来の VoD 配信の形態をとる。ユーザの視聴確率を予想するには、ユーザの視聴履歴の収集を行い、解析することでそのユーザの趣味嗜好を調査できるようにする。その嗜好情報を用いることでそのユーザが視聴するであろう映像コンテンツを予測し、不必要な配信トラフィックの増加を防ぐことが可能となる。なお、ユーザ視聴履歴の解析と視聴確率の予想は今後の課題である。

上記のユーザの趣味嗜好を考慮した配信をレコメンド配信と定義する。レコメンド配信によって、ユーザ毎にオススメコンテンツを設定することができ、ロングテール映像コンテンツ配信が可能となる。

4. システム基本設計

本章では、提案方式を実現するためのシステム基本設計について述べる。

4.1 基本機能

本提案方式は、VoD システムであることから、基本的に STB からのダウンロード要求により、サーバから STB へコンテンツがダウンロードされることでコンテンツ配信が実現される。本方式が特徴的なのは、全ての STB に対して、どのコンテンツをいつ配信（配信はダウンロードにより実現）するのかを、サーバ側で事前に決定することができることである。このことにより、コンテンツをそのコンテンツの公開日（公開時刻）よりも事前に指定の STB へ配信させることが可能となるばかりではなく、サーバへの同時アクセス集中をできるだけ計画的に分散させることも可能となる。

このためには、本システムは次の機能を有する必要がある。

(1)STB によるコンテンツダウンロードがタイムテーブルに従って実行される機能

ダウンロード要求は STB 側から出されるものであるため、STB にはダウンロード処理がタイムテーブルに従って開始されるようにする。

(2)VoD システムであることから、ユーザのその時に見たいコンテンツ視聴要求の実現を邪魔しない機能

ユーザのオンデマンド視聴要求にも支障がないようにすべきである。また、回線速度には限りがあることから、ユーザのオンデマンド視聴要求によるダウンロードが実行されている間

は、そうではないダウンロードを一時停止するなどして、ネットワーク負荷の上昇を抑える必要が発生する。そのため、ダウンロードの目的により複数のダウンロードタイプを定義し、各ダウンロードタイプの間でダウンロード実行の優先度に差をつける。

(3)各 STB へ STB ごとに個別のダウンロード実行情報を配信する機能

STB のダウンロード実行情報については、何らかの方法でサーバから各 STB へ配布する必要がある。これについては、STB がサーバから定期的にダウンロード実行情報を取得するようにする。

(4)各 STB 個別のダウンロード実行情報をサーバ側で生成する機能

STB でコンテンツダウンロードを管理するタイムテーブルが、全ての STB で同一の場合はサーバへの同時アクセスが集中してしまう。そのため、STB ごとにダウンロード実行情報を個別に生成する必要がある。

(5)STB へダウンロードされたコンテンツの視聴可否が、時刻情報を利用して管理する機能

コンテンツごとに視聴可能期間データを設定することで、STB へダウンロードされたコンテンツはいつでも視聴可能というわけではなく、指定された期間にならないと視聴ができない。この機能があることで、公開日前配信が可能となる。

4.2 ダウンロードタイプの定義

本システムでは、目的別に以下の3種類のダウンロードタイプを定義する。

即時型ダウンロード これは、ユーザのオンデマンド視聴を実現するためのダウンロードタイプである。ユーザの STB 操作によりオンデマンド視聴要求が出されたときに実行されるダウンロードタイプである。3種類のダウンロードタイプの中では、ダウンロード実行の優先度が最も高い。即時型ダウンロード開始後に、全てのコンテンツデータがダウンロード完了しないうちに、ユーザがそのコンテンツの視聴をやめたしまった場合は、そのダウンロードタイプは予約型ダウンロードへ変化する。

PUSH 型ダウンロード これは、コンテンツのユーザ公開日前に、サーバ側で決められたユーザに対して、強制的に配信するためのダウンロードタイプである。サーバへのダウンロード要求がサーバへ同時に集中しないように、各 STB のダウンロード実行情報が生成されている。3種類のダウンロードタイプの中では、即時型ダウンロードの次に優先度が高い。複数の PUSH 型ダウンロードタイプが存在する場合は、ダウンロード開始時刻順に優先度が高い。本ダウンロードタイプは、あらかじめ多数の視聴が予想される人気コンテンツの配信や、特定のユーザへ薦めるコマーシャル映像の配信などへの使用を念頭においている。

予約型ダウンロード これは、上記2つ以外のダウンロードを目的としたダウンロードタイプである。ユーザが後で見られるかも知れないコンテンツのダウンロードなどへの使用を念頭においている。3種類のダウンロードタイプの中では最も優先度が低い。複数の予約型ダウンロードが存在する場合には、ダウンロード要求の発生順に優先度が高い。

4.3 ダウンロードコネクション数

1台の STB とサーバ間のコンテンツダウンロードコネクショ

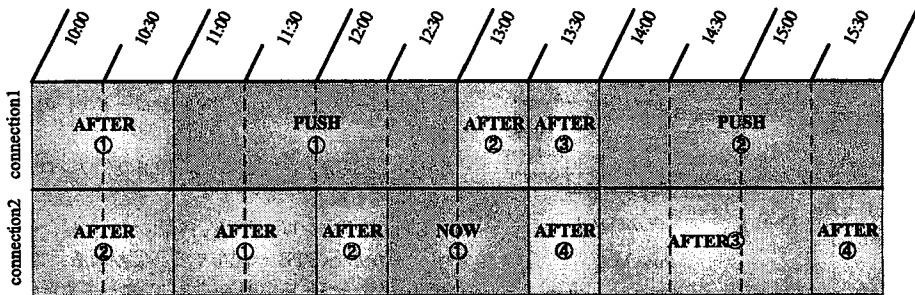


図2 スケジュール例

ン数を複数を経ることができ、最低一つを即時型ダウンロード用とし、最低一つを PUSH 型ダウンロード用とし（共に使用していないときは、他のダウンロードタイプでも利用可能）、このことを前提として PUSH 型ダウンロードの実行情報をサーバ側で生成することで、どのタイミングで即時型ダウンロードの要求が発生しても、公開日前配信用のコンテンツ配信スケジュールへの影響がないことになる。本システムでは、通信帯域が無制限ではないインターネットでの利用を念頭においていることから、そのコネクション数に制限を設けている。本稿では、コネクション数の上限を 2 とする。

4.4 コンテンツリスト

STB のダウンロード実行情報については、何らかの方法でサーバから各 STB へ配布する必要がある。本システムでは、各 STB がサーバから定期的にダウンロードによって取得するユーザ向けのコンテンツの一覧表のデータ中に、ダウンロード実行情報も添付することにより実現する。本稿では、このコンテンツの一覧表のことをコンテンツリストと定義する。

4.5 コンテンツダウンロードの動作例

本システムの STB におけるコンテンツのダウンロード動作シーケンスの一例（ある日の 10:00 から 16:00 までの間）を図 2 に示す。ここでは、一つの STB が同時にダウンロード実行可能な数を 2 としている。またここでは、即時型ダウンロードを NOW、PUSH 型ダウンロードを PUSH、予約型ダウンロードを AFTER と略す。

本動作例における想定条件

10:00 時点での STB のダウンロードリストには、PUSH 型ダウンロードが 2 件（11:00 ダウンロード開始が 1 件（PUSH①）および 13:00 ダウンロード開始が 1 件（PUSH②））、予約型ダウンロードが 3 件（登録の早い順（古い順）に AFTER①、AFTER②、AFTER③）登録されているものとする。また、12:30 にユーザが STB を操作して即時型ダウンロードを要求し、そのコンテンツの視聴を開始する。13:30 には、ユーザは STB を操作しコンテンツの視聴を途中で止めてしまうが、そのコンテンツのダウンロードはその時点で完了していないものと仮定する。ダウンロードリストとは、STB が実行すべきダウンロードのスケジュールテーブルであるが、詳細は後述する。

動作例

10:00 この時点では開始すべき PUSH 型ダウンロードが存在しないため、ダウンロードリストに登録されている予約型ダウンロード 3 件の中から最も早くダウンロードリストに登録された AFTER① およびその次に早く登録された AFTER② のダウ

ンロードを開始する。

11:00 PUSH① のダウンロード開始時刻になったが、AFTER①、AFTER② のダウンロードが実行中である。STB の同時ダウンロード実行数の上限は 2 であるため、現在実行されている AFTER①、AFTER② のうち遅くダウンロードが開始された AFTER② のダウンロードを一時停止させ、PUSH① のダウンロードを開始する。

12:00 AFTER① のダウンロードが終了した。この時点でダウンロードテーブルにあるダウンロード未完了のものは AFTER② と AFTER③ が存在するが、AFTER② の方が登録が早いことから AFTER② のダウンロードを再開する。

12:30 ユーザが STB を操作し、即時型ダウンロードを要求したため、ダウンロードリストに NOW① を追加する。すると、この時点で PUSH① のダウンロードが完了していないため、再度 AFTER② のダウンロードを一時停止する。

13:00 PUSH① のダウンロードが終了した。この時点で一番優先度の高い AFTER② のダウンロードを再開する。

13:30 ここでユーザが視聴を途中で取りやめた。しかし、そのコンテンツのダウンロードはまだ完了していないため、NOW① を予約型ダウンロードへ切り替える。この時点では予約型ダウンロードの最新は AFTER③ であることから、AFTER③ とする。また、同時に AFTER② のダウンロードが終了した。そのため、ダウンロードを実行中のものがなくなったため、AFTER③ および AFTER④ のダウンロードを開始する。

14:00 PUSH② のダウンロード開始時刻になった。実行中の AFTER③ および AFTER④ のうち、登録の新しいものは AFTER④ であるため、AFTER④ のダウンロードを一時停止し、PUSH② のダウンロードを開始する。

15:30 AFTER③ のダウンロードが終了した。そのため AFTER③ のダウンロードを再開させる。

5. プロトタイプシステムの実装

5.1 システム構成

本システムは、コンテンツ送信者である配信サーバ（Delivery Server）とコンテンツ受信者である受信端末（STB）から構成され（図 3）、インターネットを介して映像コンテンツの配信が行われる。

配信サーバは、映像コンテンツアップロードをアップロード画面表示部（Upload I/F）によって受け付ける。アップロードされた映像コンテンツはコンテンツ蓄積部（Content Storage）に蓄積され、メタデータがコンテンツリスト管理部（Content

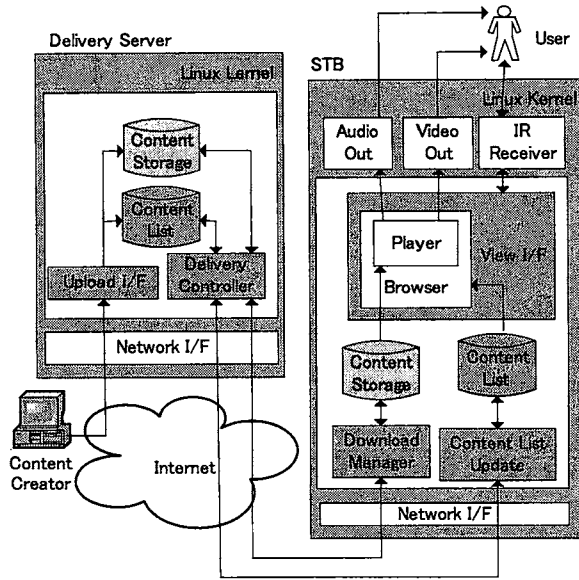


図3 システム構成

List)に登録される。配信制御部(Delivery Controller)は受信端末からのリクエストに応じて、コンテンツやコンテンツリストを配信する。

一方、受信端末については、コンテンツリスト更新部(Content List Update)は定期的にコンテンツリスト更新の有無を配信サーバに問い合わせ、更新分をコンテンツリスト管理部(Content List)に登録する。ダウンロードタスク管理部(Download Manager)は、第4節で定義した各ダウンロードタイプの実行タイミングを管理し、ダウンロードしたコンテンツをコンテンツ蓄積部(Content Storage)に保存する。ユーザ視聴画面表示部(View I/F)は、コンテンツリストからコンテンツ選択画面を生成し、ユーザが選択したコンテンツを再生する。

5.2 コンテンツリスト

本システムにおけるコンテンツリストとは、XMLで記述され、映像コンテンツのタイトル、ジャンル、キャスト、スタッフといった一般的な映像コンテンツのメタ情報の他に、ダウンロード要求送信時刻の要素を持つコンテンツ一覧である。

5.3 プッシュ型コンテンツ配信の実現方法 [8]

提案するシステムは、プッシュ型配信システムであるが、一般的なVODシステムと同様に、受信端末から配信サーバに対して配信要求を出すことで、映像コンテンツ等の情報配信が開始される。これは、現在のインターネットでは、グローバルIPを個人で持つケースは少なく、一意でないIPを特定することが難しいため、受信端末側からの配信要求なしで配信サーバ側から配信を開始することができないからである。そこで本システムでは、従来のVODサービスに加え、受信端末が配信サーバに対して受信端末の各種情報を定期的に送信することで、配信要求を発生させる方法を提案する。基本的な動作シーケンスは次のとおりである。

(1) 受信端末には、タスクマネージャが実装され、ユーザの視聴情報やコンテンツリストのバージョン情報などの受信端

表1 実装環境

OS	Linux(Kernel 2.6)
HTTPd	Apache 2.0
RDBMS	MySQL 4.1
Java	JDK 1.5
Servlet, JSP	Tomcat 5.5
Player	VLC media Player 0.8
Browser	Firefox 1.0

末情報を、定期的に配信サーバに送信する。

(2) 受信端末情報の送信が配信サーバに対するコンテンツリストの配信要求となり、コンテンツリストが配信サーバから受信端末へ配信される。

(3) コンテンツリストには、映像コンテンツのメタ情報のほかに、映像コンテンツごとのダウンロード要求送信時刻情報が含まれている。受信端末は、受信したコンテンツリストに含まれるダウンロード要求送信時刻になると自動的に該当映像コンテンツのダウンロードを実行する。

5.4 実装環境と機能動作

受信端末、配信サーバ双方でOSとしてLinuxを用いる。配信サーバではHTTPサーバを構築し、サーバレットによるWEBアプリケーションを実装する。受信端末ではJavaを用いてHTTP通信を行い、JSPによるHTMLベースのユーザインターフェイスを実装する。本システムを開発する際に使用する環境・言語を表1にまとめる。

配信サーバ・受信端末間ではHTTP通信を行う。映像コンテンツダウンロードの際のシーケンスを以下に示す。

認証にはセッション管理を用い、アクセスURLにセッションIDを付加することで、不正なアクセスを防ぐ。

受信端末 個人情報テーブルのユーザID、パスワード、MACアドレスを認証情報としてPOSTメソッドにより送信する。

配信サーバ 送られてきた認証情報と配信サーバ側の個人情報

表 2 ダウンロードリストテーブル

#	フィールド概要	フィールド名	データタイプ	備考
1	コンテンツ ID	contents_id	int(5)	
2	登録日時	regist_date	timestamp	
3	優先度	priority	int(1)	1:NOW 2:PUSH 3:AFTER
4	ステータス	status	int(1)	0:WAIT 1:DOWNLOADING 2:COMPLETE 3:PAUSE

テーブルと比較し、認証成功したらセッションを開始し、セッション ID をレスポンスする。

認証後、セッションが継続中であれば下記のように映像コンテンツのダウンロードを行う。

受信端末 セッション ID を付加した URL に対して、コンテンツ ID を GET メソッドに付加して送信する。

配信サーバ セッションが有効であるか確認し、有効であれば、指定された映像コンテンツのバイトデータを送信する。

認証後、セッションが継続中であれば下記のようにコンテンツリストの取得を行う。

受信端末 セッション ID を付加した URL に対して、コンテンツリストバージョン情報と視聴情報を POST メソッドにより送信する。

配信サーバ セッションが有効であるか確認し、有効であれば、最新バージョンとユーザの保有するバージョンの差分を XML 化し、送信する。視聴情報を配信サーバのデータベースに保存する。

5.5 配信サーバ機能

アップロード画面表示部は、JSP による WEB アプリケーションである。ここでは映像コンテンツをアップロードし、付加するメタデータを登録する。メタデータ登録時にコンテンツの公開日と PUSH 配信の実行を指定すると、ユーザ毎に個別化されたダウンロード開始時刻が生成される。

配信制御部は、Servlet を用いた WEB アプリケーションである。受信端末からのコンテンツリスト更新リクエストや映像コンテンツダウンロードリクエストを解析し、該当データを配信する。データストリーム送信メソッドをマルチスレッド化することにより複数ユーザ同時アクセスに対応する。

5.6 受信端末機能

ユーザ視聴画面表示部は、コンテンツリストからコンテンツ選択画面をローカルの HTML ページとして生成し、WEB ブラウザを用いて表示する。ユーザがコンテンツを選択するとコンテンツ再生ページを生成、表示を行う。コンテンツ再生ページには VLC media player を埋め込むことで WEB ブラウザ上でのコンテンツ再生を行う。ユーザ視聴画面では、即時ダウンロードのコンテンツをタイムシフト再生する必要がある。この機能は、VLC media player の未完全ファイル再生機能を用いて実現する [7]。

ダウンロードタスク管理部は、ダウンロードのスケジュールテーブルであるダウンロードリストを監視する。ダウンロードリストの構造を図 2 に示す。

コンテンツリスト更新部は、コンテンツリストの更新の有無を定期的にサーバに問い合わせる Java によるアプリケーションである。ここではコンテンツリストの更新を 1 時間毎としている。配信サーバの配信制御部に HTTP リクエストを送信し、レスポンスとして最新のコンテンツリストを取得する。また、コンテンツリストの更新リクエスト時にユーザの視聴履歴を配信サーバに送信することで、サーバ側でオススメコンテンツ設定時の参考とすることができる。

6. む す び

本稿では、スケジュール化された Push 型配信を可能とする VoD システムについて提案し、プロトタイプシステムの実装について詳細に説明した。提案 VoD システムによって配信サーバ負荷の平滑化、大規模一斉公開、ロングテール映像コンテンツ配信を可能とすることを述べた。以下に今後の課題についてまとめる。

- ユーザの視聴履歴の解析を行い、視聴確率を予想するアルゴリズムを策定する。また、得られた視聴確率から最適なオススメコンテンツ配信スケジュールを作成する。
- 予約ダウンロードのダウンロード完了時刻が保証されない事に関して検討する。
- HDD 空き領域が少なくなった際に、ユーザの視聴傾向による優先度を設けた映像コンテンツ削除アルゴリズムを検討する。

文 献

- [1] 総務省, "平成 19 年度版 情報通信白書," p154, 2007
- [2] Apple - iPod + iTunes -, <http://www.apple.com/itunes/music/>
- [3] Amazon.jp, <http://www.amazon.co.jp/>
- [4] 門林, 阿部, 南, "CDN を用いた次世代インターネット技術の開発・普及の試み," 情報処理学会研究報告, p59, 1-9, 2001.
- [5] 山本幹, "アプリケーションレベルマルチキャスト," 電子情報通信学会技術研究報告, CS2005-78, pp. 37-42, 2005.
- [6] 田中, 河合, 小山, 山口, 山本, "WWW における時間的制約のあるプッシュ型情報は威信機構の設計とプロトタイプ実装," 情報処理学会論文誌, Vol2005, No113, pp207-214, 2005.
- [7] VLC media player, <http://www.videolan.org/vlc/>
- [8] 知花, 応和, 池上, 高木, 富永, "プッシュ型 VOD システムのためのコンテンツリスト配送方式の検討," 情報処理学会研究報告, vol.2006, no.132, 2006-AVM-55, pp.99-102, Dec. 2006.