

オンデマンドマルチキャストネットワークを実現する コンテンツルーティングシステムの提案と実装

池田 康[†] 苗村 知明[†] 胡 天成[†] 中沢 実[†] 服部 進実[†]

[†]金沢工業大学工学部 〒921-8501 石川県石川郡野々市町扇が丘7-1

E-mail: †{ikeda, namura, tchu, nakazawa, hattori}@infor.kanazawa-it.ac.jp

あらまし

マルチキャストはインターネット上での伝送コストを減らす主な方法の1つであるが、一般的にライブ配信に用いられており、オンデマンドアプリケーションに適用される事例は少ない。これは、オンデマンドマルチキャストを実現するためには同じマルチキャストストリームに対して同時刻にアクセスする事が求められるからであり、この制約がオンデマンドマルチキャストアプリケーションの実現を難しくしている。そこで、独自に開発したコンテンツルーティングシステムを用いたオンデマンドマルチキャストを提案する。本稿では、コンテンツルーティングシステムを用いたマルチキャストビデオオンデマンドの実現手法について述べる。

キーワード マルチキャスト、コンテンツルーティングシステム、VoD

A Proposal and Implementation of the On-Demand Multicast Network Using Content Routing System

Yasushi IKEDA[†] Tomoaki NAMURA[†] Ko TENSEI[†] Minoru NAKAZAWA[†]
and Shimmi HATTORI[†]

[†]Faculty of Engineering, Kanazawa Institute of Technology

7-1 Oigigaoka, Nonoichimachi, Ishikawagun, Ishikawa, 921-8501 Japan

E-mail: †{ikeda, namura, tchu, nakazawa, hattori}@infor.kanazawa-it.ac.jp

Abstract

Multicast is one of the major methods used to reduce the delivery cost over the Internet. However, it is generally used in live broadcasting rather than in on-demand applications. That's because simultaneous access upon the same multicast stream is required in order to implement the on-demand multicast, which increases the difficulty of implementing on-demand multicast applications. Therefore, we propose a new scheme of on-demand multicast using an independently developed content routing system. In this paper, we discuss the implementation of a multicast VoD application using the content routing system.

Keyword Multicast, Content Routing System, VoD

1. はじめに

近年、インターネットのブロードバンド化に伴い WWW や E-mail などのテキストデータだけでなく、音声や動画といったマルチメディアデータを用いたコンテンツ配信サービスが行われるようになった。その中でも Video on Demand (以下、VoD) は、

ホームエンターテイメント、デジタルビデオ・ライブラリー、ムービー・オンデマンド、遠隔授業、遠隔会議、インターネットショッピング、ニュース・オンデマンド、医療情報サービスなど様々な分野での活用が期待されている。

しかし、映像等を含むマルチメディアデータはテキストデータに比べデータ容量が大きいだけでなく、連続的に安定したネットワ

ーク帯域を確保する必要があり伝送コストが高い。このため、クライアントからの配信要求が多くなるとネットワークおよびサーバの処理能力を超え、サービス遅延・停止が発生する。よって、VoDサービスを大規模かつ広域的に行うためには、何らかの手段を講じる必要がある。手段として考えられる3つを挙げ、順に説明する。

- (1) サーバおよびネットワーク設備の増強
- (2) コーデック技術の進歩
- (3) 伝送コストの低い配信アーキテクチャの開発

サーバおよびネットワーク設備の増強とは、クライアントからの配信要求数増加に対して性能や設置台数、QoS 制御などの向上によってサービス遅延・停止を回避しようというもので、例としてサーバクラスタリング技術やCDN 技術といったものがある。しかし、設備コストと適応範囲にはトレードオフの関係が存在することに注意が必要である。現在、コンテンツ配信におけるビジネスモデルが確立されているとは言いがたい状況で、回収不能な設備投資を行うことはできず、実際にかけることのできる設備コストは少ない。したがって、この方法では大規模 VoD 配信は可能ではあるが難しい。

コーデック技術の進歩とは、既存コーデック技術よりも低いビットレートで同等の映像品質が得られる技術の開発によって、配信要求数増加に備えるということである。例として MPEG4 や Windows Media 9 Series などがある。しかし、コーデック技術の進歩は音声・映像品質の向上やコンテンツの低容量化、移動端末などの狭帯域アクセスラインのためものといった意味合いの方が強く、それだけで配信要求数増加に対応できるとは言いがたい。

よって、3つ目の手段である伝送コストの低い配信アーキテクチャの開発が必要となる。ここでは、伝送コストをいかに低く抑えるかが重要な課題となるが、今回 IP マルチキャスト（以下、マルチキャスト）に着目する。マルチキャストとは、IP ネットワーク上で伝送コストを減らす手法であり、1度のデータ送信で複数のクライアントが同一データを受信する事ができる技術のことである。これによりサーバの負荷は減少し、またネットワークの帯域を効率的に利用することが可能になる。マルチキャストによる利点をまとめ、下に挙げる。

- (i) コネクション数の減少によってネットワーク帯域消費量が減少し、ネットワーク全体での負荷が減少
- (ii) コネクション数の減少によってサーバの負荷が軽減され、システム全体のスループットが向上

しかし、既存マルチキャスト配信では複数のクライアントがオンデマンドで同じマルチキャストストリームを受信するために同時にアクセスすることが求められる。ゆえに、マルチキャストはクライアントが様々な時刻にアクセスする VoD システムのようなオンデマンドマルチキャストアプリケーションには不向きとされ、IP レベルでのオンデマンドマルチキャストは普及せず、主にライブ配信に用いられてきた。

そこで、本稿ではオンデマンドマルチキャストネットワークアーキテクチャ（以下、ODMN アーキテクチャ）を提案し、その具体例として低伝送コスト配信を可能とするマルチキャスト VoD（以下、MVOD）の実現手法を述べる。

第2章で今回提案する ODMN アーキテクチャについて、第3章で MVOD とそれを実現する Content Routing System（以下、

CRS）、第4章では CRS の設計および実装、第5章にまとめと今後の予定について説明する。

2. オンデマンドマルチキャストネットワーク

アーキテクチャ

オンデマンドマルチキャストネットワークを実現するためには、複数のクライアントとサーバの間で入出力同期をとる必要がある。そこで、クライアントからサーバに対して行われるリクエストを仲介しマルチキャストセッション開始を制御するシステムによって、クライアント/サーバ間での入出力同期を行う。このシステムを仲介エージェント（以下、CRS）とよぶこととし、図1を用いて CRS の役割を説明する。

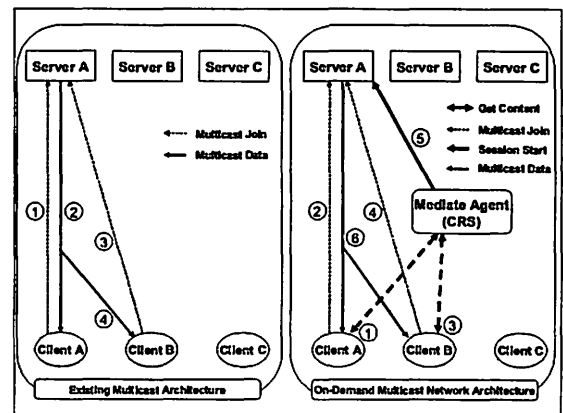


図1 オンデマンドマルチキャストネットワークアーキテクチャ

まず、既存マルチキャストアーキテクチャの動作の流れについて説明する。クライアント A がマルチキャストグループに参加するとその時点でマルチキャストデータを受信する（左図の①、②）。その後、クライアント B が同一のマルチキャストグループに参加すると、既に配信されているマルチキャストデータの途中から受信することになる（左図の③、④）。つまり、クライアント A とクライアント B がオンデマンドに同一マルチキャストデータを受信するためには、同時刻にマルチキャストグループに参加する必要がある。クライアントのリクエストは様々な時点で発生するものであり、同時刻にリクエストをさせることは難しい。これが、マルチキャストがオンデマンドアプリケーションに不向きとされる理由である。

一方、提案する ODMN アーキテクチャでは、クライアント A はまず CRS にリクエストを行い、CRS はクライアント A に対しマルチキャストグループへのリダイレクトを行う（右図の①、②）。この時点ではまだマルチキャストセッションは開始されておらず、クライアント A がマルチキャストグループに参加するだけである。続いてクライアント B が CRS にリクエストを行い、同様にマルチキャストグループへのリダイレクトが行われる（右図の③、④）。CRS はある時間待機した後、サーバ A に対してマルチキャストセッション開始を指示する（右図の⑤）。サーバ A は CRS の指示を受けてマルチキャストデータを送信する（右図の⑥）。CRS によって、クライアント A とクライアント B は同時にマルチキャストデータを受信することが可能となり、オンデマンドマルチキャストを

実現することができる。

2.1. CRS とサーバの関係

図1では、CRS からサーバに対してマルチキャストセッション開始を通知している(右図の⑤)が、オンデマンドマルチキャストを実現するためにはこの通知が必須となる。

ユニキャストではクライアントからのリクエストを動機にセッション開始を行えるが、マルチキャストの場合には、クライアントのリクエストはサーバには到達しないため、CRS からの通知がなければサーバは何時セッションを開始すればよいのかを判断できない。前述したように、もともとマルチキャストはライブ配信向けでありオンデマンド配信については想定されていないため、クライアントからのリクエストを動機にセッションを開始する仕組みがない。よって、CRS とサーバ間ではセッション開始についてのやり取りが最低限必要となる。

以上、ODMN アーキテクチャの説明をしたが、その具体例として本アーキテクチャを使用した MVOD を次に述べる。

3. マルチキャストビデオオンデマンド

大規模 VoD 配信を可能とする技術として、ODMN アーキテクチャを利用した MVOD を提案する。次に、CRS について説明する。

3.1. コンテンツルーティングシステム

CRS とは、アプリケーションレイヤでリクエストルーティングを行う独自に開発したシステムであり、バッチングを用いて MVOD を実現する[1][2]。CRS を用いた MVOD の利点を5つにまとめ下に挙げる。

(1) 低コスト化

バッチングを用いることでクライアントからのリクエスト数に対する規模拡張性が、ユニキャスト配信と比較して格段に向上する。それに伴ってサーバおよびネットワークに求められるコストを抑えることができる。結果的に大規模 VoD 配信を可能とする。

(2) システムの規模拡張性が高い

VoD サーバ数の増加に対して CRS の設定を変更すればよいだけでなく、本システムはマルチキャストプロトコルに依存しないため、VoD システムの規模拡張性が高いといえる。

(3) 部分適用性が高い

CRS にてサーバ割当てを制御できるため、マルチキャスト対応/非対応サーバが混在できる。また、クライアント情報に基づいてマルチキャスト対応/非対応サーバどちらを割当てるか切替えが可能である。

(4) クライアントに特化した機能を持てる

CRS にてクライアントに関する情報 (IP アドレス、端末情報など) を取得して、それを基にしたサーバおよびコンテンツ割当てが可能である。システムの実装方法によっては、クライアントの認証や、サービスの差別化なども行うことができる。

(5) アドレス管理の集中化

複数サーバ環境であっても、CRS にてマルチキャストアドレスを把握し集中管理ができる。ただし、システムの実装

方法によってマルチキャストアドレスの管理方法が様々なので、必ずしも利点とはいえない。

3.2. バッチング

バッチングとは MVOD 技術の一つで、ある時間 (以下、バッチング時間) に同一コンテンツに行われたコンテンツ要求を全て束ね、一つのマルチキャストストリーミングで処理する。バッチング時間中にどれだけ多くのコンテンツ要求が行われたとしても一つのマルチキャストセッションで処理できるため、ユニキャスト配信と比較してサーバ効処理しなければならないセッション数を大幅に削減でき、VoD システムが収容可能ユーザ数を向上させることができる。

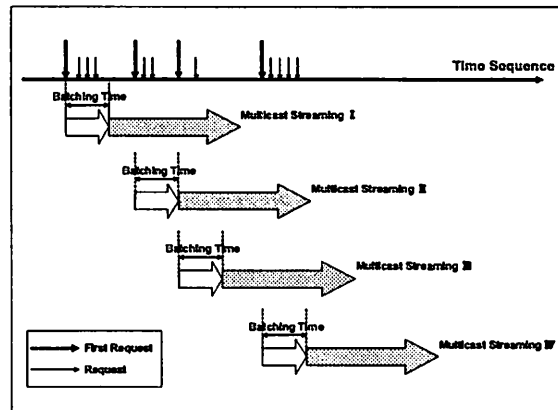


図2 バッチングの概念図

図2は、バッチングの概念を表したものである。バッチングでは、最初のコンテンツ要求 (図中の First Request) がなされてからバッチング時間後にストリーミング配信が行われる。図ではコンテンツ要求数14をマルチキャストストリーム数4で処理できている。

バッチングを用いると MVOD が実現可能であるが、クライアントに対してバッチング時間という遅延を与えるため、正確には NVOD (Near VoD) [3] とよばれる分類になる。また、バッチング時には長いほど VoD システムのスループットが向上し、短いほどクライアントに対する即時性が向上するというトレードオフが存在する。

3.3. コンテンツルーティングシステムの動作

次に、CRS の動作について説明する。動作をシーケンス図でまとめたものが図3であるが、基本的には2章で説明した ODMN アーキテクチャに基づいて動作する。ただし、図中のクライアント A、B、C はそれぞれ同一コンテンツにリクエストをしたものとする。また、クライアントの CRS への割当ては Web ページのリンクや DNS などの何らかのリクエストルーティングシステムなどによって行われたものとする。

CRS は、クライアントからのリクエストを受信すると割当てするサーバおよびコンテンツを独自アルゴリズムによって導出し、クライアントをマルチキャストグループにリダイレクトする。クライアントは CRS の指示通りにマルチキャストグループに参加する。CRS はバッチング時間が終了次第、サーバに対してマルチキャストセッション開始を促す。サーバは CRS からの通知を動機にマルチキャストストリーミング配信を行う。

一つのマルチキャストストリーミングを配信する毎に一つのマルチキャストアドレスが必要であるので、サーバおよびコンテンツの割当ては具体的にはマルチキャストグループへのリダイレクトによって行われる。

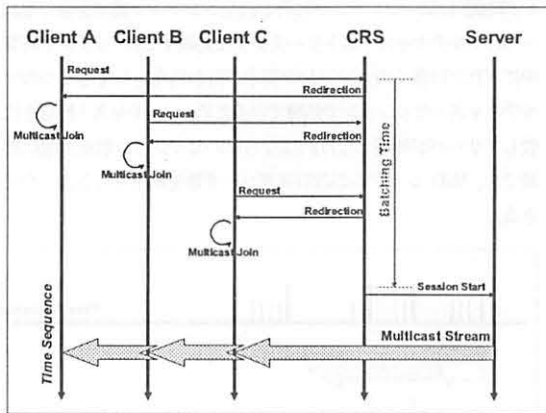


図3. コンテンツルーティングシステムの動作シーケンス図

クライアントがマルチキャストグループに参加する時の挙動や、参加した後のマルチキャストツリーの作成方法、サーバからクライアントまでのマルチキャストデータの配送のされ方などマルチキャストプロトコルと密接に関わる部分はCRSの及ぶ範囲ではない。CRSはアプリケーションレイヤで動作し、あくまでもマルチキャストを利用するシステムであることに注意が必要である。その代わり、マルチキャストプロトコルとは独立した運用ができる。

4. CRS の設計および実装

今回、ストリーミングサーバソフトウェアとして米Microsoft社のWindows Media Technologyを、CRSとサーバとの通信はSOAP (Simple Object Access Protocol) を用いて行う。これらを用いる理由は、ストリーミング再生ソフトであるWindows Media Playerの普及と、SOAPはセキュリティと今後の機能拡張を考慮した結果である。CRSアーキテクチャを概念化したものを図4に示す。

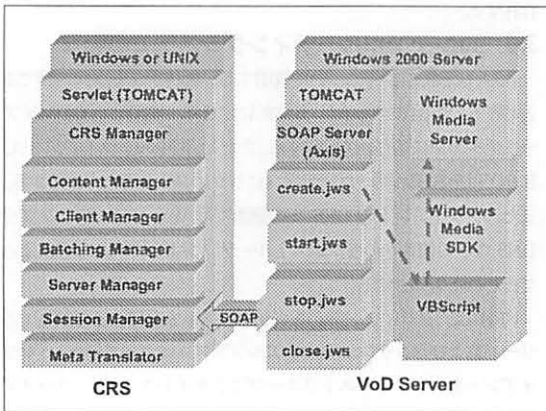


図4. CRSアーキテクチャ

4.1. CRS 全体設計

実用レベルで動作しかつ今後広範な範囲にCRSが普及できるよ

う、CRSはJavaでCRSクラスライブラリとして作成した。CRSマネージャが提供するAPIを利用できる環境ならば、どのシステム上であってもCRSは動作できる。今回は、Windows2000ServerとServlet (TOMCAT) [4]という構成を用いた。クライアントからのリクエストの受信とリダイレクトはServlet APIを用いて行う。

また、サーバOSにはWindows 2000 Server、ストリーミングサーバソフトウェアにはWindows Media Serverを、CRSと通信するプラグインはWindows Media SDKを用いて作成した。CRSとのやり取りについてはセッションマネージャのところで詳しく述べる。順に各コンポーネントの説明をする。

4.2. CRS Manager

CRSマネージャはServletを使用してクライアントからのリクエスト情報を受け取り、各マネージャを呼び出し、クライアントに対してリダイレクトを行う役割を持つ。

4.3. Content Manager

コンテンツマネージャは、コンテンツ情報の管理・操作機能を提供する。現状では、クライアントから要求のあったコンテンツがマルチキャスト処理可能/不可能を判断する機能を持つ。今後、コンテンツのメタ情報を解析し、それに基づいて配信の仕組みを変更する手法などを検討する。

4.4. Client Manager

クライアントマネージャは、クライアント情報の管理・操作機能を提供する。現状ではコンテンツ要求を行ったクライアントがマルチキャスト処理可能/不可能かを判断する機能を持つ。今後、様々なクライアント情報を用いて配信方法を変更する手法について検討する。例として、クライアントの端末情報 (マシンスペック、ユーザエージェント、ブラウザなど) やアクセスネットワーク情報などによって、配信するコンテンツを切替えたりバッチングアルゴリズムを変更したりすることが考えられる。

4.5. Server Manager

サーバマネージャは、サーバ情報の管理・操作機能を提供する。具体的な役割は、割当てるサーバを決定することである。割当て方法は大きく分けて2つあり、サーバ負荷情報を用いたサーバ選択アルゴリズムによるものと、静的割当てのものである。

前者は、サーバ負荷情報を各サーバから取得し、サーバ選択アルゴリズムを用いて解析することによって割当てるサーバを導出する。サーバ選択アルゴリズムについてはMVDODの本質的な部分ではないので本稿では触れない。

サーバマネージャには、いくつものサーバ選択アルゴリズムを組み込み可能であり、CRSマネージャでそれらを選択できる。今後は、前述したコンテンツ情報やクライアント情報なども使用してサーバ選択アルゴリズムを切替えられる方式を検討中である。

静的割当ては、あらかじめ何らかのルールに従って割当てるサーバを決めておく方法である。例として、クライアントIPに基づいた静的割当てと、ラウンドロビンがある。

4.6. Session Manager

セッションマネージャは、サーバと通信しセッションの作成・開始・削除などのセッション管理を行う。

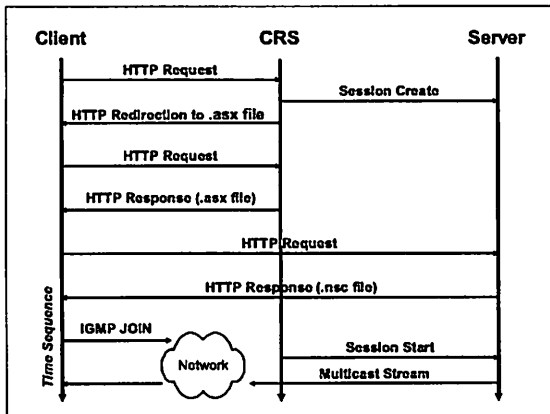


図5. シーケンス図

今回、Windows Media Server (Windows 2000 Server) をサーバとして用いるため、マルチキャストストリーミングを行う場合には、サーバ側にステーションという仮想ディレクトリパスを作成する必要があり、クライアントがストリームを受信するにはステーション情報ファイル (.nsc ファイル) をダウンロードしなければならない。ユニキャストストリーミングの場合には、ストリーミングデータ (avi ファイルや wmv ファイルなど) の URL さえわかれば、メタファイルをダウンロードする必要はないのだが、マルチキャストの場合にはステーション情報ファイルをプレイヤーに読み込まなければならない。よって、CRS とサーバおよびクライアントとのやり取りをシーケンス図で表すと図5になる。

また、CRS とサーバとのやり取りが増えたのは、ステーションはマルチキャストストリーム毎に必要なため、バッチングしたグループ個数分のステーションおよびステーション情報ファイルを作成および削除をしなければならないからである。

マルチキャストグループの割当ては、設定ファイルに記述されたマルチキャストアドレスの範囲で行われる。

4.7. Batching Manager

バッチングマネージャはバッチングの管理と実行を行う。バッチングアルゴリズムに基づいてバッチング時間の決定と実行を行うのだが、バッチングマネージャには様々なアルゴリズムを組み込み可能である。例として、サーバ負荷をパラメータにバッチング時間を決定するものや、コンテンツ人気度に応じて決定するものがある。サーバ負荷が高くなる傾向にある場合には、バッチング時間を長くすることでサーバが処理するマルチキャストストリーム数を減らし、負荷の増加を抑える。しかし、この場合一つのマルチキャストストリームで処理するクライアント数が増加するため、バッチング効率は高くなるがクライアントへの即時性は低下する。

また、バッチング時間が長くなるとクライアント側でタイムアウトが発生する可能性があるため対策が必要であるが、現在のところバッチング時間をタイムアウトが起こらない短い時間に設定している。

4.8. Meta Translator

メタトランスレータは、メタファイル (.asx ファイル) の生成を行う。メタファイルには、ストリーミングデータの实体がある URL と様々なオプション (有効期限、著作権、広告へのリンクな

ど) を記述できる。Windows Media Player では、直接ストリーミングデータの URL に対してリクエストを行うこともできるが、まずメタファイルを読み込ませてその中に記述されているストリーミングデータの URL に対してリクエストさせるのが一般的である。マルチキャストの場合には、メタファイルにステーション情報ファイルの URL を記述して図5で示した手順で処理を行う。

4.9. Plug-in

プラグインは、CRS のセッションマネージャと SOAP を用いて通信する。プラグインが提供するサービスは、ステーションの作成・開始・削除の3つである。今回は Windows Media Technology 対応のプラグインのみ作成したが、サーバアーキテクチャにあわせてプラグインを用意すれば、セッションマネージャから SOAP を用いて一元的なサービス呼び出しが可能である。

5. 実験

CRS を用いて実験および評価を行う。まず実験環境について説明し、ついで実験結果について述べる。

5.1. 実験環境

本実験は、図6で示す実験環境によって行い、実験項目は下記に示す4つである。クライアントからのリクエストは、本実験用に開発した疑似リクエスト送信プログラムによってクライアントから CRS に対して行われる。ただし、ユニキャスト VoD 用の疑似リクエスト送信プログラムの開発が間に合わなかったため、ユニキャストに関する実験結果は10クライアントによる10リクエスト分の結果について近似曲線をとったものとなっている。ユニキャストの場合、リクエストはクライアントから直接ストリーミングサーバに対して行われる。また、本実験で使用されるコンテンツは全てビットレートが314kbps (wmv 形式) で再生時間が5分以上のものである。

- (1) ユニキャスト VoD と MVD との比較
- (2) リクエスト発生率によるバッチング効果の比較
- (3) コンテンツ数によるバッチング効果の比較
- (4) コンテンツ要求頻度によるバッチング効果の比較

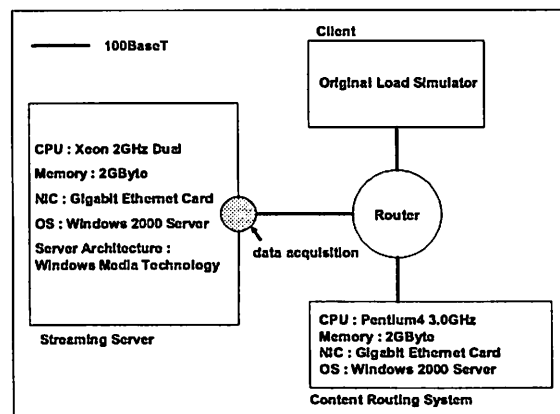


図6. 実験環境

5.2. 実験結果と評価

まず、(1)についての結果を図7に示す。ただし、本実験項目では全リクエストは単一のコンテンツになされる。ユニキャスト VoD

の場合 240 秒後必要となるビットレートが 90Mbps となるのに対し、バッティング時間 (図中の BT) が 5 秒の場合で約 18Mbps、15 秒で約 7Mbps、30 秒で約 3Mbps となる。これらの結果から、リクエストがバッティング時間に対して短い間隔で断続的に行われる場合、ネットワークトラフィックの面でユニキャスト VoD に比べ MVOD が優位である。また、バッティング時間が長いほどトラフィックを低減できることがわかる。

つづいて、(2)の結果を表したのが図 8 である。トラフィック量がリクエスト発生率に関わらないことがわかる。

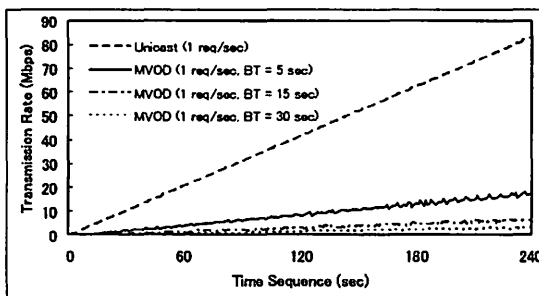


図 7. ユニキャスト VoD と MVOD との比較

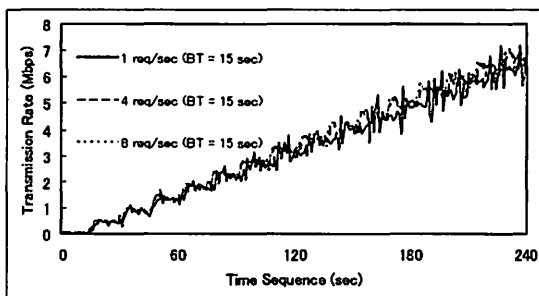


図 8. リクエスト発生率によるバッティング効果の比較

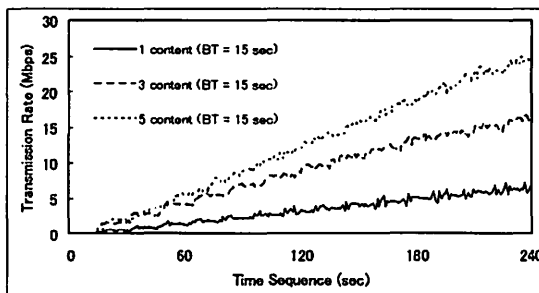


図 9. コンテンツ数によるバッティング効果の比較

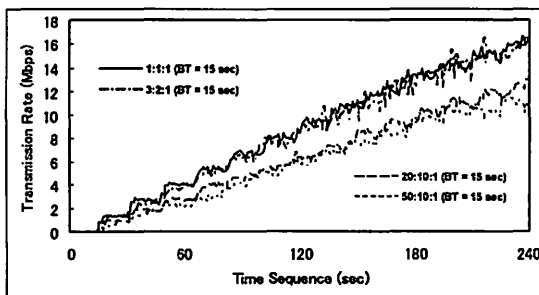


図 10. コンテンツ要求頻度によるバッティング効果の比較

言い方をかえればリクエストが密な間隔で行われるほどバッティングの効果が表れるといえる。

図 9 は(3)の結果を示したものであるが、各コンテンツに対するリクエストは一様分布に従ってランダムに 1 秒間に 1 回行われる。240 秒の時点でコンテンツ数 1、3、5 それぞれ約 6.6Mbps、約 16.3Mbps、約 24.5Mbps となった。この結果より、コンテンツ数が増加するとリクエストが分散するので、バッティング数が増えサーバが処理しななければならないストリーム数が増加、それに伴ってネットワークトラフィックが増すことがわかる。

(4)では、コンテンツ数を 3 とした時、各コンテンツに対する要求頻度が均等である場合と偏りがある場合とでネットワークトラフィックにどのような違いが出るかを評価した。結果を図 10 に示す。各コンテンツへの要求頻度は図中に示す割合でランダムに 1 秒間に 1 回行われる。図より、要求頻度が 1:1:1 の場合と 3:2:1 の場合でほとんど違いが見られないことがわかる。これは、バッティング時間に対して要求頻度の偏りが少ないためである。また、20:10:1 の場合と 50:10:1 の場合は、バッティング時間に対して要求頻度が偏っているためにネットワークトラフィックが低減されている。これらの結果から、コンテンツ要求頻度が偏っているほど、バッティングの効果が高いことがわかる。

6. まとめ

本論文では既存コンテンツ配信技術の規模拡張性についての問題点に言及し、オンデマンドマルチキャストネットワークアーキテクチャによる解決を提案した。その具体例として CRS を用いたマルチキャストビデオオンデマンドを示し、その設計および実装方法について説明を行った。CRS を用いた実験では、バッティングの効果を評価し CRS の有効性について論じた。本提案手法は、その規模拡張性の高さから大規模 VoD 配信や少ないサーバコストでの VoD 利用を可能とする技術であると考えられる。

今後としては、複数サーバ環境における CRS の評価実験や、実用化を踏まえた実証実験を行う予定である。

謝辞 本研究は、通信・放送機構の地域提案型研究開発制度の支援を受けてここに実施された。ここに記して謝意を表す。

文献

- [1] 池田康、松井優詞、胡天成、中沢実、服部進実、"ポリシ選択型 CDN プラットフォームの構成と検証"、電子情報通信学会論文誌(B), Vol.J86-B, No.3,(2003)
- [2] 池田康、胡天成、中沢実、服部進実、"コンテンツルーティングシステムを用いたマルチキャストビデオオンデマンドの提案と実装"、電子情報通信学会情報ネットワーク研究会技術報告 NS2002-267,pp43-48
- [3] Huadong Ma, Kang G Shin, "Multicast Video-on-Demand Services", in ACM SIGCOMM Computer Communication Review Volume 32, Issue 1, pp.31-43, January 2002
- [4] The Apache Jakarta Project, <http://jakarta.apache.org/tomcat/>