

ユニキャストとマルチキャストを融合させたストリーミング手法

片山貴嗣 石渡秀典 大亦寿之 市川健一郎 重野寛 岡田 謙一
慶應義塾大学理工学部

近年、ブロードバンド化が進み、ユーザが映像を視聴したい時に視聴ができる VOD (Video-On-Demand) が登場した。しかし、VOD にはサーバに過大な負荷がかかるといった問題がある。そこでサーバに負荷をかけずに放送可能な NVOD (Near-Video-On-Demand) が考えられたが、NVOD にもユーザに映像を届けるまでに遅延が生じてしまうといった問題が挙げられる。そこで本研究では、ユニキャストとマルチキャストを融合させ、サーバの負担を軽減し、ユーザの要求に対し低遅延で放送を可能とするストリーミング手法を提案する。そして、そのストリーミング手法を実装し評価を行う。

A Proposal of The Streaming System by Fusion of a Unicast and Multicast

Takashi Katayama Hidenori Ishiwata Hisayuki Ohmata
Ken-ichiro Ichikawa Hiroshi Shigeno Ken-ichi Okada
Faculty of Science and Technology,
Keio University

Recently, advance in computing technology and high-speed networking has made the delivery of on-demand video service (video-on-demand, VOD). In traditional VOD, the number of streams in a server can be high if the requirement is too much. For more requirement, the use of multicasting is more scalable and cost-effective. Such a technique is referred to as NVOD (Near-Video-On-Demand). But, in NVOD, it is possible that the user delay is high. In this research, I propose streaming technique which a unicast and multicasting are united so that the damage giving to server is reduced, broadcasting to user by low delay. I experiment and evaluate this scheme.

1 はじめに

近年、ブロードバンド化が進み、世界的にインターネットが普及している。そして、パソコンの性能およびネットワークの高速化が進み、VOD (Video-On-Demand) のサービスを実現可能となった。VOD とはユーザが視聴したいと要求した時にコンテンツを視聴できるサービスである。VOD

を実現するシステムとして、サーバがユニキャストを用いて視聴要求した各ユーザに対して配信する手法があるが、このシステムは要求が少ない映像コンテンツを配信する際には有効だが、要求の多い映像を配信する際には不適切である。そこで、視聴要求が多い映像コンテンツを配信するには、マルチキャストにより放送することが有効であると考えられ、マルチキャストを利用して多数の要

求に対応可能なサービスとして登場したのが near-VOD(NVOD)である。NVODとはある一定時間ごとにマルチキャストにより繰り返し放送する手法であるが、サーバに負荷がかからないように間隔をあけて放送すると、ユーザが視聴要求してから映像を表示するまでにかかる遅延が長くなるといった問題点が考えられる。そこで、本研究ではユニキャストとマルチキャストを融合させて、サーバに負荷をかけず、ユーザに対して低遅延で映像コンテンツを放送可能なストリーミング手法”Fusion-Stream”を提案する。これは、サーバから要求の多い映像コンテンツを放送する際に利用する手法である。放送開始時間前に視聴したいと要求した各ユーザにマルチキャストで放送し、放送開始時間に遅れて視聴要求してきたユーザに対しては、これから放送される映像コンテンツはマルチキャストグループに加入してサーバから受け取り、すでに放送済みの映像コンテンツは直前に要求した上位ユーザからユニキャストにより受信する。これによりサーバへの負荷が過大になることはなく、ユーザに対して視聴要求後すぐに映像を届けことが可能となる。以上の有効性を確認するために、「Fusion-Stream」を実装し、実験を行い評価した。第2章では、関連研究の紹介をし、第3章ではストリーミング手法を提案する。第4章にて第3章で述べたストリーミング手法を実装し、第5章で実験、評価を行う。最後に第6章で、本論文をまとめる。

2 関連研究

ユニキャストを利用して要求したクライアントに対し、サーバからユニキャストで配信するVODシステムが登場したが、このシステムでは、常に1対1通信であるため、視聴要求するユーザの数に比例して、送信するストリームの数が増加する。映像を視聴したいという要求が過大になったとき、円滑に映像を配信するために広大な帯域が必要となり、サーバへの負荷も過大となってしまう。そのため、要求の多い映像コンテンツを配信するには不適切である。そこで、マルチキャストを利用してサーバに負荷をかけずに放送可能なNVODシステムが考案されてきた。以下でNVODのシステムを紹介する。

2.1 Pyramid Broadcasting

PyramidBroadcasting[1]とは放送対象となる映像コンテンツを細かいセグメントに分割して放送するシステムである。第1セグメントを最小容量のセグメントとし、第2セグメントは第1セグメントの2倍の容量をもつセグメントとする。さらに第3セグメントは第2セグメントの容量の2倍とする。このように分割し、放送する。ユーザから視聴要求があると、そのユーザに対して、まず第1セグメントを配信する。そして、ユーザはその第1セグメントをある程度バッファに溜め込み視聴する。そしてユーザが第1セグメントを視聴している間に、第2セグメントを配信し、バッファに溜め込む。PyramidBroadcastingとは、この放送を連続的に行うストリーミング手法である。そして第1セグメントの容量を小さくすることにより、ユーザが視聴するまでの時間が短く設定可能である点が、このシステムの利点である。しかし、ユーザは映像コンテンツの約70%のバッファ領域を必要とする。つまり、配信する映像コンテンツの容量が大きくなるにしたがって、必要なバッファ領域も増加するということになる。これがこの手法の最大の問題点であると考えられる。

2.2 Harmonic Broadcasting

HarmonicBroadcasting[2]も放送対象となる映像コンテンツを細かいセグメントに分割して放送するシステムである。まず、映像コンテンツをN個に等分割し、S1、S2、S3・・・とする。次にS2をS2.1、S2.2と2つに等分割、S3をS3.1、S3.2、S3.3と3つに等分割する。このようにS1、S2・・・とだんだん容量が小さくなるように分割する。そして、ユーザから視聴要求があると、サーバはS1、S2.1、S3.1・・・とまとめて放送し、ユーザはある程度バッファに溜め込み視聴を開始する。そして、ユーザがS1、S2.1と視聴している間にS2.2、S3.2、S4.2・・・と放送する。そして、S2.1を視聴し終わるとユーザはS2.2、S3.1、S3.2とバッファから読み出し視聴する。このようにバッファに溜め込みながら視聴する。しかし、映像コンテンツをS1、S2.1、S2.2、S3.1・・・のように正しい順番に並べ替えて視聴しなくてはならないといった複

雑さが、このシステムの最大の問題点といえる。この他にも [3][4][5][6][7][8][9][10] などの NVOD システムがあるがいずれも同様な問題が挙げられる。

3 ストリーミング手法の提案

第2章では、NVODのシステムを紹介し、それらにおける問題点について述べた。3章では、サーバから多くのユーザに対して映像コンテンツを放送する際に考えられるサーバへの過大な負荷、ユーザが映像を視聴できるまでにかかる遅延といった問題を改善すべく、新たなストリーミング手法”Fusion-Stream”を提案する。

この手法は要求数の多い映像を放送する場合に用い、サーバから映像コンテンツを放送し、放送開始時間から遅れて視聴要求してきたクライアントに対しても、放送開始時間から視聴しているクライアント同様に最初から視聴できるものである。

放送開始時間前から視聴したいと要求したクライアントをグループ化して、サーバからマルチキャスト放送する。そして、遅れて要求したクライアントは近くのネットワークですでに視聴しているクライアントからユニキャストにより映像コンテンツを受信する。(図1)

このように、”Fusion-Stream”とはユニキャストとマルチキャストを利用してサーバへ過大な負荷をかけず、ユーザに低遅延で映像を配信するシステムである。第4章で本システムの詳細を述べる。

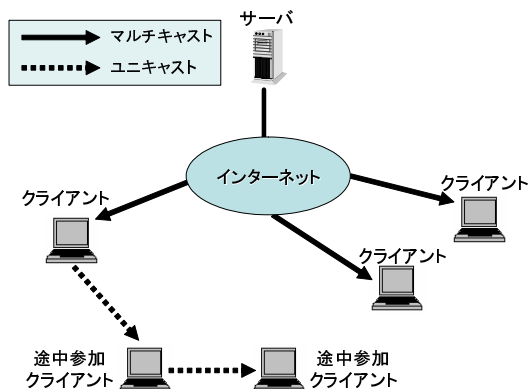


図1: Fusion-Streamの全体像
fig.1: Outline of Fusion-Stream

4 Fusion-Streamの実装

第3章において提案した”Fusion-Stream”はサーバから多数のクライアントに対して映像コンテンツを配信するためのシステムである。本章ではサーバ、クライアントの機能について説明する。

4.1 Server

3人のクライアントが視聴したいとサーバに要求した時のネットワーク状況を図2、そのときのサーバが保持するリスト内の状態を図3に示した。まず、サーバからの放送開始時間前にクライアントA、Bが視聴要求をサーバにしたとする。サーバは図3の $i1_i$ 、 $i2_i$ にクライアントA、BのIPアドレスを要求した順に書き込む。そして、クライアントA、Bをマルチキャストグループに加入し映像コンテンツを放送する(図2の1のネットワーク状況)。次に放送開始時間後にクライアントCがサーバに視聴要求をしたとする。サーバはリストの[3]にクライアントCのIPアドレスを書き込み、クライアントCをマルチキャストグループに加入する。また、クライアントCの直前に加入したクライアントBのIPアドレス([2]に書き込まれているIPアドレス)をクライアントCに送信し、クライアントBにクライアントCのIPアドレスを送信する。

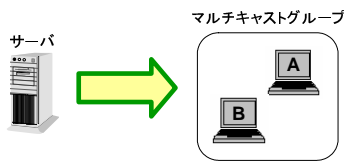
クライアントCに対しては既にサーバから放送済みの映像コンテンツをクライアントBからユニキャストで配信し、視聴要求後の映像コンテンツはサーバからマルチキャスト放送する。

4.2 Client

サーバからの映像放送時間前に視聴要求をしたクライアントは、マルチキャストグループに加入する。そして、放送開始時間になるとサーバは映像コンテンツをマルチキャスト放送する。クライアントは受信したコンテンツをバッファに溜め込み、同時に視聴を開始する。

次にサーバからの映像放送開始時間後に視聴したいと要求したクライアントは、サーバから自分の直前に視聴を開始した上位クライアントのIP

1. 放送開始時間直後のネットワーク状況



2. 放送開始時間後、クライアントCが視聴要求した後のネットワーク状況

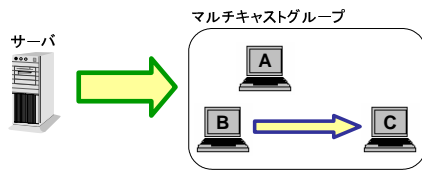
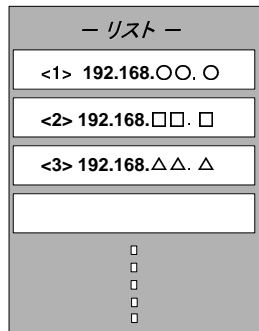


図 2: 映像放送後のネットワーク状況
fig.2: Network Condition after Broadcasting



• 視聴したいと要求したクライアントは[1]から順番にリストに書き込まれる

図 3: サーバが保持するリスト
fig.3: List of Server

アドレスとマルチキャスト放送に使用している IP アドレスを受信する。そして、既にサーバから放送済みの映像コンテンツを上位クライアントからユニキャストで受信し、視聴要求した時間からの映像コンテンツはサーバからのマルチキャスト放送により受信する。このときのクライアントが持つバッファの状態を図 4 に示した。

放送開始時間からマルチキャストグループに所属しているクライアントは最初からマルチキャスト放送により映像コンテンツを受信しバッファに溜め込み視聴する。そして、放送開始時間後に視聴要求したクライアントは放送済みの映像コンテンツを上位クライアントからユニキャストで受信し、バッファに溜め込み視聴する。また、マルチキャスト放送により受信した視聴要求時間後から

の映像コンテンツもバッファに溜め込み、ユニキャストで受信した映像コンテンツを視聴した後に視聴する。

上位クライアントは下位クライアントに映像コンテンツを配信した後、そのバッファ内のデータは削除する。

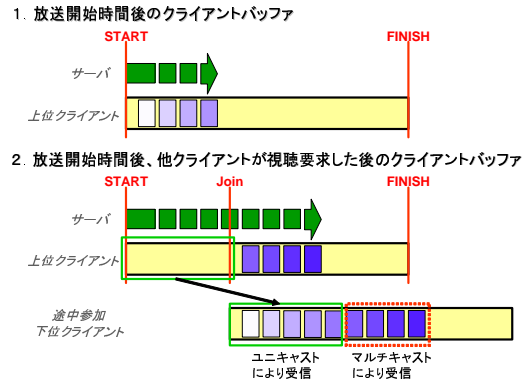


図 4: サーバから放送開始された後のクライアントバッファ

fig.4: Client Buffer

5 Fusion-Stream の実験、評価

比較対象システムを 2 つ作成し、Fusion-Stream を含め 3 つのシステムで実験し、評価を行う。

5.1 比較対象システム

- システム 1：サーバからの放送開始時間に遅れて視聴したいと要求したクライアントに対してはサーバからユニキャストで配信するシステム。
- システム 2：放送開始時間に遅れて視聴したいと要求したクライアントに対しては、直前に要求したクライアントから全映像コンテンツをユニキャストで配信するシステム。

5.2 実験環境

図 5 のようにサーバからスイッチングハブ 2 つを通して、多数のクライアントと接続する。最初

にクライアント1人がサーバに映像視聴を要求する。そしてサーバから、映像コンテンツを放送開始し、その後5秒間隔でクライアントがサーバに視聴要求をする。今回の実験では50人のクライアントにより実験した。

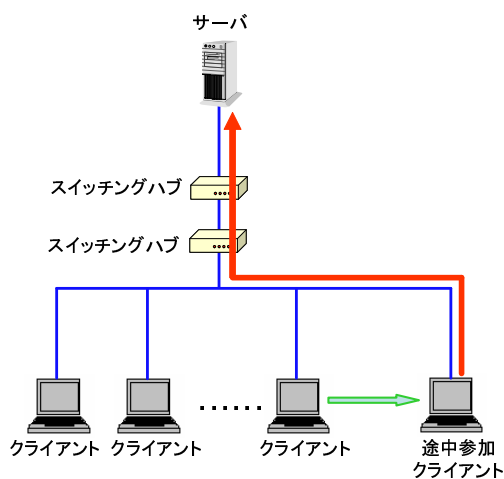


図 5: 実験環境

fig.5: Implementation Environment

5.3 評価方法

放送開始後にクライアントが視聴してから映像が表示されるまでの時間を算出し比較する(図6)。

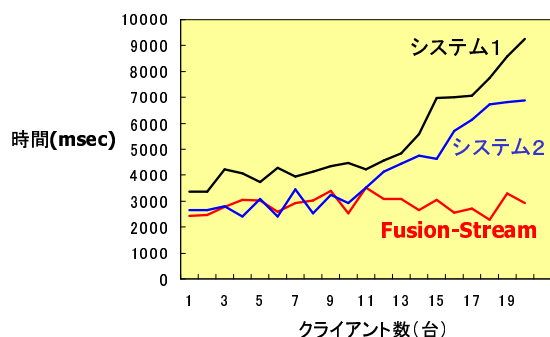


図 6: クライアントが視聴要求してから映像が表示されるまでの時間

fig.6: The Comparison of Starting Time

5.4 結果、考察

評価の結果を図6に示した。この表から全てのクライアントにおいてFusion-Streamを利用した時に、映像が表示されるまでの時間が短いということがわかる。

そして、Fusion-Streamはサーバから送るストリームは少ないことから、サーバへ過大な負荷がかからないシステムであるといえる。

さらに、システム1、システム2と比較して、Fusion-Streamは放送開始後、時間が経過するとネットワーク内に存在するストリーム数が一定になる。"Fusion-Stream"はサーバからの放送開始時間後に視聴要求したクライアントは、視聴要求した後の映像コンテンツをマルチキャストグループに加入することで受信するシステムであり、より安定していると言える。

以上のことから、Fusion-Streamを利用することで、サーバから映像コンテンツを放送する際、多数のユーザが視聴要求をしてきた場合サーバに過大な負荷がかかり、ユーザが視聴要求してから映像を表示するまでにかなりの遅延が発生してしまうという2つの問題点が解決されたことを確認した。

6 結論

本研究では、「ユニキャスト」と「マルチキャスト」を融合させたストリーミング手法" Fusion-Stream"を提案した。Fusion-Streamとは、サーバから要求の多い映像コンテンツを放送する際に利用する手法である。放送開始時間前に視聴したいと要求した各ユーザにマルチキャストで放送し、放送開始時間に遅れて視聴要求してきたユーザに対しては、これから放送される映像コンテンツはマルチキャストグループに加入してサーバから受け取り、すでに放送済みの映像コンテンツは直前に要求した上位ユーザからユニキャストにより受信する。これによりサーバへの負荷が過大になることはなく、ユーザに対して視聴要求後すぐに映像を届けることが可能となる。以上の有効性を確認するために、「Fusion-Stream」を実装し、実験を行い評価した。

今後の課題として、多チャンネル放送への対応、

さらに多数のクライアントへ放送可能などといったシステムを構想し、MulticastVOD のさらなる研究を行っていく。

謝辞

本研究は科学研究費・基盤研究 (C) の研究助成によるものである。ここに記して謝意を表す。

参考文献

- [1] S.Viswanathan and T.Imielinski. Metropolitan area video-on-demand service using pyramid broadcasting. *Multimedia Syst.*, Vol. 4, pp. 197–208, Sept 1996.
- [2] L.-S.Juhn and L.-M.Tseng. Harmonic broadcasting for video-on-demand service. *IEEE Trans.Broadcasting*, Vol. 43, pp. 268–271, Sept 1997.
- [3] Simon Sheu Kien A.Hua. Skyscraper broadcasting: A new broadcasting scheme for metropolitan video-on-demand systems. *ACM Computer Commun.Rev.*, Vol. 27, pp. 89–100, Oct 1997.
- [4] E.L.Abram-Profeta and Kang G.Shin. Scheduling videl programs in near video-on-demand systems. *Proc.ACM Multimedia'97*, pp. 359–369, Nov 1997.
- [5] S.W.Carter J.-F.Paris and D.D.E.Long. A universal distribution protocol for video-on-demand. *in Proc.Int.Conf.Multimedia and Expo*, pp. 49–52, July 30-Aug.2 2000.
- [6] A.Fox et al. Adapting to network and client variation viaon-demand,dynamic distillation. *Proc.ACM ASPLOS-VII*, pp. 160–170, Oct 1996.
- [7] S. W. Carter and D. D. E. Long. Improving video-on-demand server efficiency through stream tapping. *Proc. IEEE ICCCN ' 97*, pp. 200–207, Sept 1997.
- [8] A.L. Chervenak and D.A. Patterson. Choosing the best storage system for video server. *Proc. of ACM Multimedia ' 95*, pp. 109–119, 1995.
- [9] M. Ferris D.L. Eager and M.K. Vernon. Optimized regional caching for on-demand data delivery. *Proc. Multimedia Computing and Networking Conference (MMCN ' 99)*, 1999.
- [10] Z.-L. Zhang L. Gao and D. Towsley. Catching and selective catching: efficient latency reduction techniques for delivering continuous multimedia streams. *Proc. ACM multimedia ' 99*, pp. 203–206, Nov 1999.