Vol.2011-GN-78 No.16
2011/1/22

記念写真撮影機能を備えた インターネット放送システムの実装と評価

齊藤義仰† 村山優子†

インターネット放送における動画品質は，テレビの品質にはいまだ追いついていない．視聴者は高品質な動画を望むが，動画配信による膨大な通信トラフィックにより動画品質の向上は難しい．本稿では，視聴者のユーザ体験向上を目指し，高品質な写真撮影機能を備えたインターネット放送システムを提案する．動画配信には通信トラフィックを抑えるため品質を下げた動画を利用する一方で，視聴者にいつでも好きなシーンの高品質な写真を撮影できる機能を提供することで，ネットワークに負荷を与えずに視聴者のユーザ体験を向上させる．写真撮影機能がどのように視聴者によって利用されるのかを検証するため，プロトタイプシステムの設計と実装を行った．また，プロトタイプシステムを用いた岩手県立大学大学祭での実験結果について報告する．

Implementation and Evaluation of an Interactive Internet Live Broadcasting System with a High-Quality Snapshot Function

Yoshia Saito† and Yuko Murayama†

The video quality on the Internet does not come up to TV quality yet. Although audience often ask for high-quality videos, it is difficult to improve the quality because of its huge video traffic. In this paper, we propose an Internet broadcast system with a high-quality snapshot function to improve user experience. While the proposed system delivers low-quality video to audience, it provides a high-quality snapshot function which enables the audience to take a snapshot of a desired and favorite scene anytime. To assess effectiveness of the snapshot function, we designed and implemented a prototype system. This paper reports results of an experiment in our university festival using the prototype system.

1. はじめに *

近年，インターネット回線の高速化により，ウェブ上で動画コンテンツを視聴することが可能になってきた．動画プレーヤはウェブブラウザに統合され，インターネットユーザは特別なクライアントソフトウェアをインストールすることなく，容易に動画を視聴できる．このような Adobe Flash や Microsoft Silverlight 等の動画再生プラットフォームがウェブ上での動画視聴を一般化させた． YouTube[1]に代表される動画共有サービスには，数千万の動画が共有され，様々な国々のインターネットユーザからオンデマンド動画が視聴されている．さらに近年，インターネットユーザによるライブ動画配信も人気を集めている．多くのインターネットユーザが PC とウェブカメラを用いて，個人的なライブ動画配信を行っている．今後，動画共有サービスやライブ動画配信サービス（これらをインターネット放送サービスと呼ぶものとする）はさらに拡大していき，通信トラフィックがさらに増加していくと考えられる．

これらの動画配信による膨大な通信トラフィックは，ネットワークに負荷を与え，通信費の増加を引き起こす．多くのインターネット放送サービスは広告により収益を得ているが，通信費の増加により利益を上げることが難しくなっている[2]．さらに，現在のインターネット放送サービスは数百 kbps の品質で動画を配信しているが，テレビの品質にはまだ追いついていない．インターネット放送サービスは高品質な動画を配信すべきであるが，通信費などの問題により通信トラフィックを増やすことは現状難しい．

視聴者のユーザ体験を向上させる手法は 2 つ考えられる．1 つ目は上述した通り動画の解像度やビットレートを上げることで動画品質を向上させる手法である．2 つ目は，動画にインタラクティブ機能を組み込むことで付加価値をつける手法がある．現状では後者の手法が現実的であると言える．インタラクティブ TV（iTV）[3-6]は，ユーザ体験向上のためにインタラクティブ機能を視聴者に提供する研究分野である．Ustream[7]と Justin.tv[8]はライブ動画配信に iTV の技術を取り入れた代表的なサービスである．これらのサービスでは，視聴者はライブ動画を見ながら，文字チャットやツイッター等を用いて放送者や他の視聴者とソーシャルコミュニケーションを行うことができる．インタラクティブ機能により低品質な動画であっても魅力ある放送を実現できると考えられている．

一方，我々はこれまでインターネット放送の研究をしてきた[9, 10]．実験では 100Mbps のインターネット回線を用いて数百 kbps の動画品質でライブ動画配信を行っていた．そのため，視聴者が満足できる動画を配信することが難しく，通信トラフ

* † 岩手県立大学
Iwate Prefectural University

ⓒ 2011 Information Processing Society of Japan

ィックを増やすことなくユーザ体験を向上させる機能をインターネット放送システムに組み込むことが必要であった．

そこで我々は，視聴者がライブ配信動画を見ながら好きなタイミングで高品質な写真を撮影できるインターネット放送システム，Photographable TV を提案する．提案システムを用いることで，あたかも放送会場にいて写真撮影しているかのようなユーザ体験を提供することができる．写真のデータサイズは動画と比べてはるかに小さく，提案システムは通信トラフィックを増加させることなくユーザ体験を向上させることができる．本稿では，写真撮影機能が視聴者によってどのように利用されるかを検証するため，プロトタイプシステムの設計と実装を行う．また，実装したプロトタイプシステムを用いて岩手県立大学の大学祭で実験を行い，写真撮影機能の検証を行い，実用化に向けた検討を行う．

以降 2 節では，関連研究を挙げながら提案システムの新規性について述べる．3 節では提案システムのモデルを示し，4 節ではその設計と実装について述べる．5 節ではプロトタイプシステムを用いた実験とその結果について述べ，6 節で結論と今後の課題をまとめる．

2. 関連研究

通信トラフィックを増やさずに動画品質を向上させる手法としては，IP マルチキャスト[11, 12]がある．IP マルチキャストでは，送信者は 1 つのデータストリームのみを配信し，経路上の IP マルチキャスト対応ルータが必要に応じてデータストリームの複製を行い，マルチキャストグループに属する複数の受信者にデータストリームを届ける．IP マルチキャストは経路上のルータが全て IP マルチキャストに対応していなくてはならず，インターネットを介した通信は難しく，現在のインターネット放送サービスではユニキャストが一般的に用いられている．このような理由により，実用的なシステムを実現するためにユニキャストによる配信を考える必要がある．

送信者の通信トラフィックを削減する他の手法として，P2P 技術を用いて通信トラフィックを分散させる研究[13-16]が行われている．これらの研究では， P2P ネットワーク上のホスト間で配信データの分散を行い，送信者への通信トラフィック集中を防いでいる．しかし，P2P 技術はファイアウォールなどのネットワーク構造を無視して通信を行うことが可能なため，多くのネットワークで利用が禁止されている．また，クライアントに専用ソフトウェアのインストールや設定が必要な場合が多い．そのため，一般のインターネットユーザが利用するには敷居が高い．多くのインターネットユーザに利用してもらうためには，様々なネットワークで使用することができ，専用ソフトウェアのインストールや設定を必要としないシステムが求められる．

写真撮影機能を組み込んだ類似システムとして，Chalk Talks[17]がある．Chalk Talkss は遠隔講義システムで，聴講者が黒板の内容を詳細に見ることができるように，一定時間毎に高解像度な黒板の写真を撮影して聴講者に配信する．Chalk Talk は黒板の内容を見ることが目的なため，一定期間毎の写真撮影で十分であるが，提案システムでは任意のタイミングで視聴者が見たいものが映った瞬間の写真を撮影できる必要がある点が異なる．

3. Photographable TV

Photographable TV は，視聴者に写真撮影機能を提供することで，ユーザ体験の向上を実現するインタラクティブなインターネット放送システムである．視聴者は任意のタイミングで好きなシーンの高品質な写真を撮影し，保存することができる．高品質な写真は，過剰な間隔で要求されなければ，通信トラフィックをそれほど増加させることはない．また，動画品質がそれほど高くなくてもユーザ体験を向上でき，撮影した写真を用いてソーシャルコミュニケーションを活発にできると考えられる．

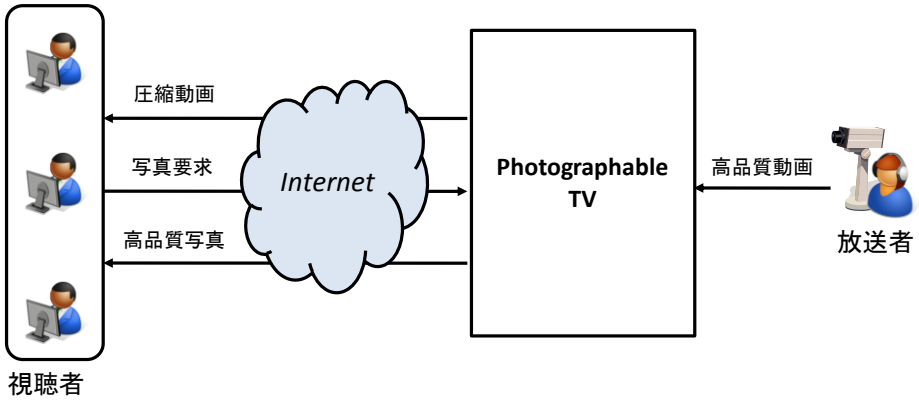

図 1　Photographable TV のシステムモデル

図 1 に提案システムである Photographable TV のシステムモデルを示す．本モデルは放送者，視聴者と提案システムにより構成される．まず始めに，放送者は高品質動画を提案システムに送信する．次に提案システムは動画を圧縮して，品質を落として視聴者に配信する．視聴者は配信された動画を見ながら，任意のタイミングで写真要求を提案システムに送信することができる．提案システムは写真要求を受け取ると，品質を落とす前の高品質動画から，高品質写真を生成し視聴者に提供する．視聴者は撮

ⓒ 2011 Information Processing Society of Japan

影された写真を確認し，保存することができる．

Photographable TV を実現するためにはいくつかの考慮すべき課題がある．まず，Photographable TV は動画エンコードと写真エンコードを行うが，生成される写真品質はオリジナルの動画品質と同等であるため，高品質な写真を生成するにオリジナル動画は非圧縮かつ高解像度でなければならない．しかし，非圧縮かつ高解像度な動画は，インターネットを介して通信するためにはデータサイズが大きすぎる．そのため，動画エンコードと写真エンコードの機能はインターネットを介さなくてもよいように，放送者の近くにある必要がある．動画エンコードと写真エンコードを放送者の所有する同一 PC 上で行うことを考えると，視聴者からの写真要求が多くなるにつれ負荷が大きくなり，処理が追いつかなくなるという問題がある．他に考慮すべき問題としては，視聴者が頻繁に写真を要求してきた場合である．Photographable TV は，写真による通信トラフィックが動画による通信トラフィックよりもかなり小さいという前提で成り立つインターネット放送システムである．しかし，視聴者が頻繁に写真を要求してくると，写真による通信トラフィックが増加するため，ネットワークキャパシティを超えないように写真による通信トラフィックを制御しなくてはいけない．システムを設計する上で，これら 2 つの点が必要条件となる．

4. プロトタイプシステムの設計と実装

写真撮影機能がどのように視聴者によって利用されるのか，また提案システムの可用性を検証するため，プロトタイプシステムの開発を行った．本節では，プロトタイプシステムの設計と実装について述べる．

4.1 システム設計

図 2 にプロトタイプシステムのシステムアーキテクチャを示す．プロトタイプシステムは，エンコードサーバ，動画配信サーバ，クライアント管理サーバ，写真管理サーバの 4 つのサーバとクライアントによって構成される．

エンコードサーバは，動画エンコード機能と写真エンコード機能を持つ．動画エンコード機能はカメラから非圧縮動画を受け取ると，配信用に動画を圧縮する．同時に，非圧縮動画は写真エンコード機能にも受け渡される．写真エンコード機能は受け取った非圧縮動画から高品質写真を生成する．

動画エンコード機能により生成された圧縮動画は，動画配信サーバに渡される．動画配信サーバは受け取った圧縮動画を，視聴要求のあったクライアントにインターネットを介してユニキャストで配信する．

クライアントは，動画配信サーバから配信された圧縮動画を GUI 上に表示する．GUI はウェブブラウザ上で動作するものとし，特別なソフトウェアのインストールなしで利用できるものとする．また，視聴者は，動画を見ながら GUI を用いて写真要求を任意のタイミングで送信することができる．

クライアント管理サーバは，動画を視聴しているクライアントからの写真要求を受け付ける．写真要求をクライアントから受け取ると，エンコードサーバ上の写真エンコード機能に写真要求を転送する．写真エンコード機能から写真準備完了メッセージを受け取ると，対応するクライアントにそのメッセージを転送する．

写真管理サーバは，写真エンコード機能から受け取った写真データの管理を行う．写真エンコード機能から写真データを受け取ると，サムネイルを生成し，写真とサムネイルを保存する．写真準備完了メッセージには，対応する写真とサムネイルの場所が記述されており，クライアントは写真管理サーバにアクセスし，まずサムネイルをダウンロードし GUI 上に表示する．視聴者は，サムネイルを選択することにより，高品質な写真をダウンロードすることができる．サムネイルを用いることにより，満足のいく写真が撮影できなかった場合の写真ダウンロードを抑制することができると考えられる．

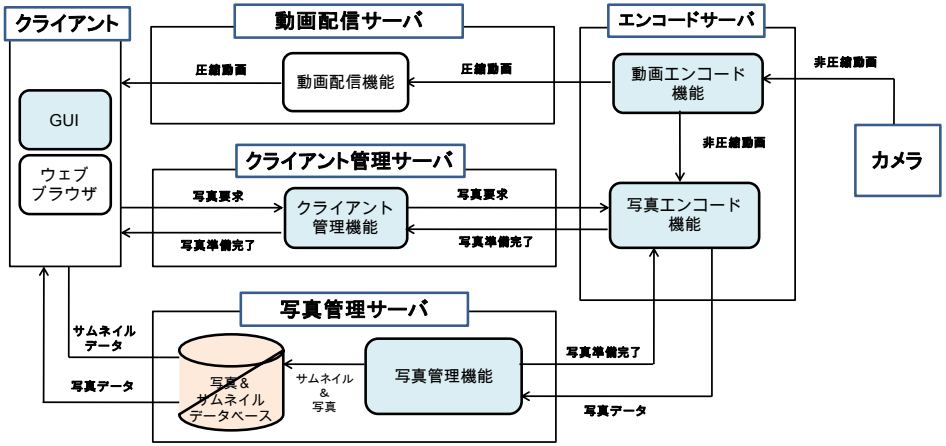

図 2 プロトタイプシステムのシステムアーキテクチャ

ⓒ 2011 Information Processing Society of Japan

4.2 システム実装

システム設計に基づき，システム実装を行った．動画エンコード機能の動画圧縮には Adobe Flash Media Live Encoder 3 を利用した．その他の動画エンコード機能，写真エンコード機能と GUI は，インターネット放送サービスで一般的に利用されている Adobe Flash で実装を行った．クライアント管理機能は様々な環境で動作できるように JAVA を用いて実装した．写真管理機能は写真へのアクセスが容易にできるようにするため，ウェブサーバ上で動作する PHP スクリプトで実装した．動画配信サーバの動画配信機能は Adobe Flash Media Server を用いた．なお，本実装では動画配信サーバとクライアント管理サーバは同一 PC 上で動作させるものとした．

4.2.1 動画エンコード機能

動画エンコード機能はカメラから，720x480 の解像度の非圧縮動画データを受け取る．非圧縮動画データは，Adobe Flash Media Live Encoder 3 によりリアルタイムで圧縮され，低解像度でビットレートを落とした圧縮動画になる．圧縮動画は数百 kbps で動画配信サーバに送信される．また，非圧縮動画の各フレームは BMP format で抽出され，写真エンコード機能に渡される．

4.2.2 写真エンコード機能

動画エンコード機能から BMP データは圧縮されていないため，720x480 の解像度の BMP データサイズは約 1MB となる．これはインターネットを介して写真管理サーバに送信するには大きすぎる．そのため，写真エンコード機能では，BMP format の写真データを JPEG format に圧縮を行う．圧縮を行うことで，約 50KB 程度のデータサイズとなり，エンコードサーバと写真管理サーバ間のトラフィックを削減することができる．

一方で，JPEG に圧縮する際に，エンコードサーバの CPU 負荷が高くなる．よって，クライアントが頻繁に写真要求をエンコードサーバに送ると，CPU 負荷が高くなりすぎ，写真生成を行えなくなってしまう問題が起こる．さらに，JPEG 圧縮して写真のデータサイズを小さくしたとしても，このような頻繁な写真要求は，エンコードサーバと写真管理サーバ間に膨大な通信トラフィックを発生させてしまう．これらの問題を解決するためにバッファリング手法を写真エンコード機能に組み込みを行った．

写真エンコード機能は BMP データを一定期間ごとにバッファに格納する．本実装では，エンコードサーバのスペックを考慮し 500ms 毎とした．写真エンコード機能は，写真要求が届くと，バッファの中から写真要求が送信されたタイミングに最も近い BMP データを検索する．この時，クライアントでの動画再生バッファ長も考慮にいれ，視聴者が撮影したいと考えているシーンと実際に撮影されるシーンの誤差が最も小さくなるように調整を行う．バッファから最も近い BMP データを発見した後，JPEG への圧縮を行い，写真管理サーバにアップロードし，ダウンロード用 URL を取得する．この際，すでに JPEG に圧縮され，写真管理サーバに送信されたデータであった場合は，以前取得したダウンロード用 URL を再利用する．このように，写真データの効率的な再利用を行えるようにすることで，サーバ負荷と通信トラフィックの削減を実現する．

4.2.3 写真管理機能

圧縮された写真は，写真管理サーバにアップロードされる．写真管理機能は，写真を受け取ると，120x90 の解像度のサムネイルを生成する．写真とサムネイルの組は，ローカルのウェブサーバ上の公開ディレクトリに保存される．今回のプロトタイプシステムでは写真を見るためのアクセス制御を行っていないが，将来的に導入する予定である．

4.2.4 クライアント管理機能

クライアント管理のために，1) クライアント ID 管理機能，2) 接続管理機能，3) 写真要求転送機能，4) ログ機能という 4 つの機能を実装した．クライアント ID 管理機能は，クライアント管理サーバにアクセスがあったクライアント毎に一意の ID を提供する．この ID を用いて接続管理機能は各クライアントの識別と管理を行う．写真要求転送機能は，クライアントからの写真要求をクライアント ID 付きで写真エンコード機能に転送を行い，クライアント ID に基づいて結果を要求があったクライアントへ返す．ログ機能は，どのクライアントがいつ何をしたのかがわかるように，接続・切断時間や写真要求を受け取った時間をクライアント ID と一緒に記録する．

4.2.5 GUI

図 3 にクライアント用 GUI を示す．圧縮された動画は上部に表示される．カメラアイコンを押すと写真要求がクライアント管理サーバに送信される．カメラアイコンがクリックされると，クライアント管理サーバから応答があるまでクリックできない．写真要求が成功すると，まずサムネイルが下部にリスト表示される．視聴者は，期待通りの写真が撮影できているようであれば，サムネイルをクリックすることで 720x680 の写真をダウンロードして見ることができる．写真はブラウザ上の別ウィンドウで表示され，視聴者のローカルディスクに保存することが可能である．

ⓒ 2011 Information Processing Society of Japan

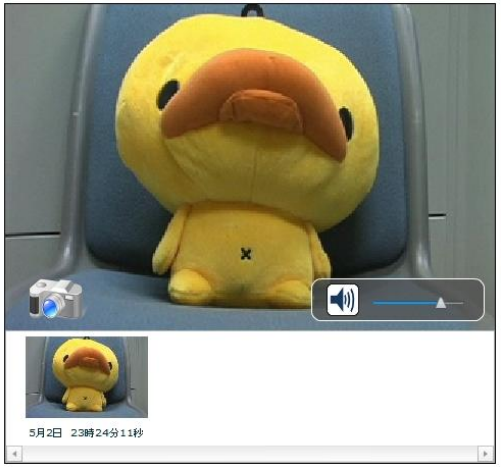

図 3　クライアント用 GUI

5. 実験

写真撮影機能がどのように利用されるのか，またシステムの可用性を検証するため，実装したプロトタイプシステムを用いて岩手県立大学の大学祭でインターネット放送実験を行った．本節では，実験の概要とその結果について述べる．

5.1 実験環境

図 4 に実験環境を示す．動画配信サーバ，クライアント管理サーバと写真管理サーバは岩手県立大学内に設置し，大学のネットワークに接続した．これらのサーバをインターネットと 100Mbps の回線で接続した．大学祭は大学敷地内で行われ，今回の放送実験では主にステージの様子をインターネット放送した．エンコードサーバをノート PC 上で動作させてステージ前に配置し，ビデオカメラを接続してステージの様子を撮影した．エンコードサーバは大学のネットワークを経由して各サーバにアクセスできるようにした．動画はビットレートを 200kbps，解像度を 320x240，フレームレート 15fps として配信した．クライアント用 GUI をインターネット上に公開し，インターネットユーザは誰でもリアルタイムに動画を視聴と，写真撮影をできるようにした．検証のため，視聴者数とエンコードサーバにおける CPU 使用率，また写真要求メッセージの到着時間を記録した．大学祭は 10/23, 24 の 2 日にわたり開催され，23 日は 10:00～17:00，24 日は 10:00～19:00 まで放送を行った．

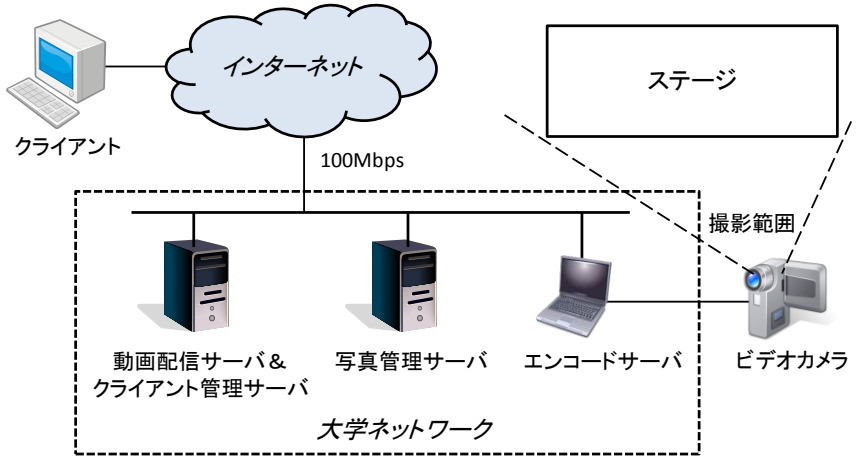

図 4　実験環境

5.2 実験結果

実験結果を分析し，どの程度の視聴者がプロトタイプシステムにアクセスし写真撮影機能を利用したのか，また写真撮影機能を利用した場合のスケーラビリティを検証した．表 1 に実験結果を示す．なお，実験結果からは実験関係者からのアクセスを除外してある．表 1 からわかるように，10/23 は約 170 人，10/24 は約 190 人の視聴者からアクセスがあった．秒単位での視聴者数を見てみると 10/23 は平均 13 人，最高 21 人からのアクセスがあり，10/24 は平均 11，最高 25 人からのアクセスがあった．実験としては十分な視聴者数を確保できたと言える．また，写真撮影機能をつかった回数を見てみると，10/23 は約 200 回，10/24 は約 360 回と写真撮影機能は多く利用されており，写真撮影の需要があることがわかった．

表 1　実験結果

	2010/10/23	2010/10/24
総視聴者数	168	193
秒単位の平均視聴者数	13	11
秒単位の最高視聴者数	21	25
写真撮影機能利用回数	202	364

さらに，CPU 使用率を分析したところ，およそ 30%～70%の範囲で遷移していた．

ⓒ 2011 Information Processing Society of Japan

写真エンコード機能に組み込んだバッファリング手法により, 写真撮影機能が CPU リソースを専有することはなかった.

6. おわりに

本稿では，ユーザ体験の向上を目指し，写真撮影機能を実現したインタラクティブなインターネット放送システムを提案した．また，プロトタイプシステムの設計と実装を行い，大学祭でインターネット放送実験を行った．インターネット放送実験の結果，写真撮影機能の需要とシステムの可用性が確認できた．

今後の課題として，視聴者数がさらに増加した場合のシステムの可用性を検証する必要がある．また，写真撮影機能によりユーザ体験がどの程度向上するのかを評価していきたい．

参考文献

1) YouTube, http://www.youtube.com/
2) Huang, C., Li, J. and Ross, W. K., “Can Internet Video-on-Demand be Profitable”, SIGCOMM'07, pp. 133-144, 2007.
3) Jensen, J., “Interactive Television - A Brief Media History”, In Proc. of EuroITV2008, pp. 1-10, 2008.
4) Cesar, P. and Chorianopoulos, K., “Interactivity and user participation in the television lifecycle: creating, sharing, and controlling content”, In Proc. of UXTV2008, pp. 125-128, 2008.
5) Ursu, M. et al, “Interactive TV narratives: Opportunities, progress, and challenges”, ACM TOMCCAP, Vol. 4, Issue 4, 2008.
6) Metcalf, C. et al., “Examining presence and lightweight messaging in a social television experience”, ACM TOMCCAP, Vol. 4, Issue 4, 2008.
7) Ustream, http://www.ustream.tv/
8) Justin.tv, http://www.justin.tv/
9) Takamizawa, N. and Murayama, Y., “Use of Telepresence in Informal Broadcasting over the Internet, In Proc. of the ACM SIGMM 2004 Workshop on Effective Telepresence”, pp. 12-15, 2004.
10) Saito Y. and Murayama, Y., “A Proposal of an Interactive Broadcasting System for Audience-driven Live TV on the Internet”, Journal of Information Processing, Vol. 18 pp. 26-37, 2010.
11) Deering, S.: Host Extensions for IP Multicasting, RFC 112, 1989
12) Eriksson, H., “MBONE: The Multicast Backbone”, Communications of the ACM, Vol. 37, No. 8, pp. 54—60, 1994.
13) Tran, D., Hua, K. and Do, T., “Zigzag: an efficient peer-to-peer scheme for media streaming”, IEEE INFOCOM 2003, pp. 1283-1292, 2003.
14) Guo, M. and Ammar, M.H., “Scalable live video streaming to cooperative clients using time shifting and video patching”, IEEE INFOCOM 2004, pp. 15001-1511, 2004.
15) Cui, Y., Li, B., and Nahrstedt, K., “oStream: Asynchronous streaming multicast in application layer overlay networks”, IEEE JSAC, Special Issue on Recent Advances in Service Overlays, Vol. 22(1), pp.91-106, 2004.
16) Qu, C., Li, B. and Zhao, S., “Exploring large-scale peer-to-peer live streaming topologies, Transactions on Multimedia Computing”, Communications, and Applications (TOMCCAP), Vol. 4, Issue 3, 2008.
17) Ichimura, S., “Delivering Chalk Talks on the Internet”, 40th Hawaii International Conference on System Sciences (HICSS), Collaboration Systems Track, IEEE CS Press, p.4-11 , 2007.

ⓒ 2011 Information Processing Society of Japan