

推薦論文

## メディアストリーミングにおける高速移動通信網に適した動的符号化レート制御手法

廣 本 正 之<sup>†1</sup> 筒 井 弘<sup>†2</sup> 越 智 裕 之<sup>†1</sup>  
小 佐 野 智 之<sup>†3</sup> 石 川 憲 洋<sup>†3</sup> 中 村 行 宏<sup>†4</sup>

高速移動通信網の発展により、携帯電話等の移動通信端末においてもネットワーク経由での動画視聴が可能となってきた。しかし移動通信網では電波の受信状態の変化等により通信帯域の大幅な変動や一時的な切断が発生するため、固定ビットレートでの配信では画像の乱れ等の発生や再生自体が停止する可能性がある。本研究ではこれらを回避するため、通信品質の変化に応じ動画ビットレートの動的制御を行い、不安定なネットワーク上でも途切れることなく視聴し続けることが可能なメディアストリーミングシステムを提案する。提案システムでは、サーバ側で実時間に対するトランスコード処理の遅延を検出することによりネットワークの変動を感知し、それに適したレート制御を行う。本研究では特に高速移動通信網を想定し、発生しうる複数の帯域変動パターンに対し提案手法の評価を行い、画像の途切れ等のない安定したストリーミングが行えるレート制御手法の検討を行った。その結果、急激な帯域変動に対しても迅速にレート制御を行い、連続したストリーミングが行えることを示した。

### Media Streaming System with Dynamic Rate Control for High Speed Mobile Networks

MASAYUKI HIROMOTO,<sup>†1</sup> HIROSHI TSUTSUI,<sup>†2</sup>  
HIROYUKI OCHI,<sup>†1</sup> TOMOYUKI OSANO,<sup>†3</sup>  
NORIHIRO ISHIKAWA<sup>†3</sup> and YUKIHIRO NAKAMURA<sup>†4</sup>

Emerging high-speed mobile networks enable us to receive media streaming data by mobile terminals on demand. However, media streaming with fixed bitrate causes some problems such as annoying artificial noise, interruption, and so forth, since link quality fluctuates dramatically in such mobile networks. Motivated by this, in this paper, we propose a novel media streaming system with

dynamic rate control scheme to achieve continuous streaming on such unstable networks. The proposed rate control scheme uses the delay of the transcoding process to estimate network bandwidth, and adjusts the bitrate of media streams dynamically. To demonstrate our system and scheme, the proposed streaming system is evaluated under some typical models of bandwidth change in high-speed mobile networks. The results show that stable and high-quality streaming without interruption can be achieved by the proposed scheme.

### 1. 序 論

近年、携帯電話等の移動通信端末におけるデータ伝送速度の向上により、携帯機器上でのメディアコンテンツのリアルタイムストリーミングが可能となってきた。また、Peer to Peer (P2P) 技術等により、デジタル機器や家電製品を相互接続しホームネットワークを形成する事例が提案されてきており、ホームネットワーク上のメディアコンテンツを P2P 技術を用いて携帯端末からストリーミング再生するシステム<sup>1)</sup> 等の開発も行われている。このような用途において、携帯端末で高品質なメディアストリーミングが行えることが今後ますます必要になると予想される。

動画等のメディアストリーミングにおいては、これまでは端末の性能に応じ、固定ビットレートで符号化されたコンテンツを配信する方式が主流であった。しかし実際のネットワークでは通信品質が必ずしも安定しているとは限らないため、通信帯域の変動によりパケットの損失や遅延等が発生し、画像の乱れや再生自体が停止するといった現象が生じる可能性がある。特に高速移動通信網を想定した場合、端末の移動にともなう電波状況の変化や回線の混雑度等により固定ネットワークに比べ通信帯域がより激しく変動するため、安定したストリーミングを行うことが困難であると考えられる。

このような問題に対し、固定ビットレートで動画を配信するのではなく、通信帯域の変

<sup>†1</sup> 京都大学大学院情報学研究科

Graduate School of Informatics, Kyoto University

<sup>†2</sup> 大阪大学大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

<sup>†3</sup> 株式会社 NTT ドコモサービス&ソリューション開発部

Service & Solution Development Department, NTT DOCOMO, INC.

<sup>†4</sup> 立命館大学総合理工学研究機構

Research Organization of Science and Engineering, Ritsumeikan University

本論文の内容は 2008 年 7 月のマルチメディア、分散、協調とモバイル (DICOMO2008) シンポジウムにて報告され、同プログラム委員長により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である。

動に応じて動画の符号化レートを適応的に変化させる，動的レート制御ストリーミング手法が考えられる．通信帯域が減少した場合は，配信対象のコンテンツのビットレートを下げることにより再生の中断，停止を回避し，逆に通信帯域が増加した場合にはコンテンツのビットレートを上げることにより動画の品質を高めることが可能となる．これにより，固定ビットレートでのストリーミングより主観画質を向上させることができると考えられる．

動的レート制御を実現するためには一般的なストリーミングシステムに加え，現在の通信帯域を推定する機構，およびそれに応じてコンテンツのビットレートをリアルタイムに変更する機構が必要となる．また，推定された帯域の情報をもとに適切なビットレートを算出するアルゴリズムについて，再生の途切れを防ぎ，かつできるだけ高画質を維持したストリーミングを可能とするための工夫を行う必要がある．動的レート制御手法はこれまでも提案されているが<sup>2)-5)</sup>，いずれも固定ネットワークを対象としており，より大きな帯域変動や一時的な切断が起こりうる高速移動通信網には適さない．そこで本研究では，通信帯域の変動が激しい携帯端末での高品位なストリーミングの実現を目指し，動的符号化レート制御を行うストリーミングシステムの提案を行う．

提案手法の特徴は次のとおりである．1 つ目は，通信帯域の急激な変動に対応するためサーバ側に動画のトランスコードを設置し，そのトランスコード処理の遅延時間によって動画の量子化ステップを制御することで動的レート制御を行う点である．既存手法 5) においてはトランスコードでのフレームレートにより帯域推定を行い，その結果によりレート制御を行っているが，提案手法ではトランスコードの遅延時間を直接用いることでより迅速なレート制御を可能としている．また，既存手法 6) では同じくトランスコードの遅延時間を利用してはいるが，遅延時間に応じ単純にフレームスキップを行っているだけであり，主観画質を著しく損なうという欠点がある．提案手法では量子化ステップを適応的に変化させることでフレームレートを保ったまま高品位なストリーミングを実現する．2 つ目は，上記のレート制御をすべてサーバ側のアプリケーション層のみで実現している点である．これにより，既存の移動通信網に特別な機能を追加することなく，また，下位層の実装に大きく依存することなく，レート制御を実現することが可能となる．かつ，レート制御をサーバ側のみで行うためクライアント側にも特別な機能を要求しないといった利点もある．

本稿ではまず 2 章で動的レート制御の概要について述べ，高速移動通信網への適用について検討する．次に 3 章で，提案する動的レート制御を行うストリーミングシステムについて述べ，4 章で高速移動通信網を想定した複数の帯域変動パターンを対象に提案システムの評価検討を行う．最後に 5 章でまとめとする．

## 2. 動的レート制御ストリーミング

動画の符号化レートを動的に制御するためには，通常のストリーミングシステムに加え通信路帯域の推定および符号化レートの制御を行う機構が必要である．本章では本研究で想定するストリーミングシステムについて述べ，動的レート制御を行うための帯域推定，符号化レート制御について述べる．また，高速移動通信網への適用についても検討する．

### 2.1 想定システム

一般的なストリーミングシステムはコンテンツを配信するサーバとそれを受信し再生するクライアントからなり，これらがネットワークを介して接続されている．動的レート制御を行うためには上記のシステムに加え，配信対象のコンテンツを要求されるビットレートに変更するための仕組みがサーバ側に必要となる．文献 2), 5), 6) ではトランスコードを用い動的レート制御を実現している．一方，文献 3), 4) ではスケラブルな動画符号化方式をそれぞれ独自に使用しレート制御を行っている．いずれの方式も有効であるが，本研究ではクライアント側に特別なデコーダが不要な，トランスコードを用いる前者の手法を採用する．トランスコードを含めたストリーミングシステムの概要を図 1 に示す．

サーバ側では，配信対象の動画の元データ (Media Source) がトランスコードによりあるビットレートへと変換され，ストリーミングサーバにより UDP もしくは TCP ネットワーク上へと送信される．ストリーミング用のプロトコルとしては Real Time Streaming Protocol (RTSP), Real-time Transport Protocol (RTP) および Real-time Transport Control Protocol (RTCP) を利用する．RTSP はストリーミングセッションの制御に，RTP はストリームデータの転送に，RTCP は RTP によるデータ転送の制御に，それぞれ用いられる．これらのプロトコルは UDP/TCP の上位プロトコルとして機能する．ネットワークへの送信時には一定量のバッファリングが行われるものとし，トランスコードおよびストリーミングサーバはつねに送信バッファを満たすように動作するものとする．送信バッファが満杯の場合は空きができるまでトランスコードは動作を停止する．

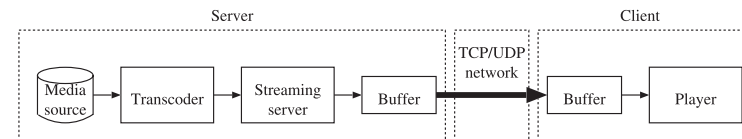


図 1 ストリーミングシステムの概要

Fig. 1 An overview of the assumed streaming system.

クライアント側ではパケットの遅延等を許容するために一定量のバッファリングを行い、メディアプレーヤにより順次動画の再生を行う。

## 2.2 帯域推定

動的レート制御を行うためには現在のネットワーク帯域を知る必要がある。物理的なネットワーク帯域はネットワークデバイスの統計情報等を観測することにより測定可能であるが、これにはネットワークプロトコルに応じたヘッダによるオーバーヘッドや、同時に稼動している他のアプリケーションによるトラフィックも含まれる。このような方法では実際にストリーミングに利用可能なネットワーク帯域を直接測定することはできないため、サーバあるいはクライアント側のアプリケーションで取得可能な何らかの情報をもとに帯域推定を行う必要がある。ストリーミングを行う際に用いられるデータ転送プロトコルである RTP の下位プロトコルとしては現在 UDP と TCP が広く利用されているが、性質の違いにより帯域推定の手法も異なってくる。以下、それぞれのネットワークにおいて帯域推定を行う方法について述べる。

### 2.2.1 UDP における帯域推定

UDP はパケットの到達確認や再送等を行わないプロトコルであり、データの信頼度を必要としない通信に用いられる。UDP は TCP と比較するとプロトコルのオーバーヘッドが少ないため高速なデータ転送が可能であるが、不安定な通信路ではパケットの欠落が起りやすいため動画等の配信においては画像の乱れの原因となる。

UDP においてはパケット到達の保証がないため、送信されたデータ量と受信されたデータ量は必ずしも一致しない。サーバ側においてはつねに所定のビットレートでパケットを送信するため、サーバ側で通信帯域を知ることは困難である。RTP/RTCP の機能によりクライアント側の統計情報を取得することも可能であるが、これはクライアント側で帯域推定を行うことと等価である。よって UDP ではクライアント側で帯域推定を行う必要があるといえる。クライアント側で帯域推定を行うための情報としては、クライアント側のバッファ使用量がしばしば用いられる<sup>7)</sup>。バッファ使用量の減少は帯域の減少によるものであるため、その変動から帯域を知ることが可能である。ただしクライアント側では帯域にどの程度余裕があるのかを知ることができないため、UDP においては通信帯域の増加を検知することは不可能である。

### 2.2.2 TCP における帯域推定

TCP はハンドシェイク方式によるコネクション型通信を行うプロトコルであり、パケットの再送や輻輳制御等の機能を持つ。TCP ではパケットの到達が保証される反面、プロト

コルのオーバーヘッドが大きく、また狭帯域の場合は輻輳制御やパケット再送等による遅延が増大するためリアルタイム性を確保するのが困難である。TCP 上での動画配信を行う場合、パケットの到達が保証されるためデータの欠損による画像の乱れは防止できるが、パケット遅延を許容できるだけの十分なバッファがクライアント側に必要となる。

TCP においては遅延があるもののパケットの到達が保証されるため、転送されたデータ量はサーバ側、クライアント側のどちらにおいても知ることができる。したがって、最終的なレート制御はサーバ側のトランスコーダで行うため TCP ではサーバ側で帯域推定を行うことが妥当である。また、帯域変動に対する追従性という観点でも、クライアント側での帯域推定はサーバ側へのフィードバックに時間を要するため、制御対象のトランスコーダを含むサーバ側での帯域推定が適すと考えられる。

図 1 のシステムを想定すると、通信路の帯域変動はサーバ側におけるバッファ使用量の変化として現れる。つまり、通信帯域の増加はバッファ使用量の減少として、帯域の減少はバッファのオーバーフローとして観測可能である。ここで、トランスコーダおよびストリーミングサーバはつねに送信バッファを満たすように動作することを考えると、通信帯域の変化はトランスコーダの動作速度として観測できることが分かる<sup>5),8)</sup>。よってトランスコーダが処理対象の動画のフレームレートと比較し、どの程度の速度で処理が行えているかにより、通信帯域が十分であるか否かが判断可能である。なお、このようにして観測される通信帯域は、ネットワーク変動による実際の帯域の変化と、それにとまなう輻輳制御やフロー制御といった TCP の送出制御による実効帯域の変化の両方を合わせたものとなる。よって、上記のようにアプリケーション層から帯域推定を行いそれをういたレート制御を行うことで、TCP の送出制御による通信帯域の変化にも対応することが可能となる。

## 2.3 符号化レートの制御

帯域推定により得られたネットワークの帯域情報をもとに、動画のビットレートを制御する。

レート制御を実現するためには、トランスコーダにおいてストリーミングの途中でビットレートを変更する機構が必要となる。動画のビットレートを制御する方法としては、解像度<sup>3)</sup>、フレームレート<sup>6)</sup>、量子化ステップの変更<sup>2),5)</sup>等があげられる。これらのうち、解像度およびフレームレートはストリームシーケンスの途中での変更が困難であるが、量子化ステップはフレーム単位で変更することが可能である。量子化ステップはトランスコード後のビットレートに直接影響するため、これを適応的に制御することによりストリーミングを中断することなく動的レート制御が可能となる。

上記のレート変更機能を持つトランスコーダを用い、推定されたネットワーク帯域をもとにビットレートの制御を行う。まず帯域が減少した場合は、クライアント側でのバッファが枯渇する前に送信ビットレートを迅速に低下させる必要がある。逆に帯域が増加した場合はビットレートの引き上げが可能となるが、急激な、あるいは早急なレートの引き上げはビットレートの振動を招く恐れがあるため、通信帯域の変動具合を考慮した緩やかな引き上げが望ましいと考えられる。これらを考慮し、ストリーミングを途切れさせることなく、できるだけ高ビットレートを保つことができる制御アルゴリズムを検討する必要がある。

#### 2.4 高速移動通信網への適用

上述のようなストリーミングシステムを構築することにより、動的レート制御を行うことが可能となる。同様の研究はこれまでも行われており、動的レート制御により主観画質が向上する結果が得られている<sup>5),8)</sup>。しかしこれらの既存研究は固定端末におけるネットワークを対象としており、帯域変動の頻度や変動幅が比較的小さい場合が仮定されている。移動通信においては固定ネットワークに比べ帯域変動が大きく、また一時的に電波が遮断されること等による通信の切断も頻繁に起こるため、このような現象も考慮する必要がある。Leiら<sup>6)</sup>は無線ネットワークを対象に、シミュレーションにより動的レート制御手法の評価を行っているが、上記のような実ネットワークにおいて起こりうる切断に関しては考慮されていない。

本研究では上記の問題に対応するため、高速移動通信網に適した動的符号化レート制御アルゴリズムの検討を行う。まず、移動通信網においては通信帯域の変動が大きく急激な帯域減少が起こりうるため、そのような場合においても動画像を途切れなく再生できるように符号化レートを迅速に低下させる必要がある。逆に符号化レートの増加については、帯域を有効利用するためにも迅速にビットレートを増加させることが望ましいが、ネットワーク帯域を超えるような過剰な増加は避けるべきである。また、電波状況の悪化や基地局の切換え等により頻繁に発生すると考えられる一時的なネットワーク切断に対応するためには、クライアント側のバッファ量を十分に確保することが重要である。バッファ量はネットワークの切断期間や特性に応じ、動画像が途切れることなく再生可能な適切な値に設定する必要がある。

以下、本稿では上記の項目について検討を行い、高速移動通信網に適した動的符号化レート制御を行うストリーミングシステムの提案を行う。

### 3. 提案システム

前章の議論をもとに、本研究で提案する動的レート制御ストリーミングシステムについて述べる。

#### 3.1 全体構成

提案システムの全体構成を図2に示す。基本的には図1と同様であり、サーバおよびクライアントがネットワークを介して接続される構成となる。サーバはトランスコーダとストリーミングサーバから、クライアントはメディアプレーヤからそれぞれ構成される。動的レート制御を行うための機構はトランスコーダに接続されている。

##### 3.1.1 ネットワークプロトコル

サーバ、クライアント間のデータ転送に用いられる RTP の下位プロトコルとしては TCP あるいは UDP を利用することが考えられるが、本研究では TCP のみを対象とする。その理由は次のとおりである。UDP は TCP に比べデータのオーバーヘッドが小さいため、高速な伝送やリアルタイム性が要求される用途において有効である。しかし UDP では前章で述べたように再送制御が行われなためパケットロスによる画像の乱れが生じやすいことに加え、プロトコルの性質上、受信状況からネットワーク帯域の増加を検知できないことがあげられる。一方、TCP においては伝送速度が遅い反面、パケットの到達が保証される。よって本研究の目指す、途切れないストリーミングを実現するためには下位プロトコルとして TCP を利用することが適すると考えられる。また、TCP ではサーバ側で帯域の変動を観測することも利点としてあげられる。なお、本研究で対象とする TCP は、TCP/IP としてインターネットで使用されている RFC 793<sup>9)</sup> 準拠のものとするが、提案手法ではアプリケーション層でレート制御を行うために TCP の実装に非依存であり、他のバージョンも使

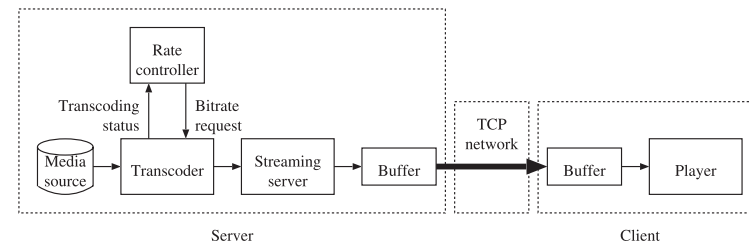


図2 提案ストリーミングシステム

Fig. 2 An overview of the proposed streaming system with dynamic rate control.

用可能である．

TCP を用いることによりサーバ側で帯域推定およびレート制御が可能となるため，本システムではサーバ側のみにレート制御機構を持たせることにする．

### 3.1.2 サーバ

ストリーミングサーバはトランスコーダにより生成されたビットストリームをそのまま配信するのみであり，一般的なストリーミングサーバが使用できる．ただし，TCP ネットワークの挙動をトランスコーダに伝えるために，出力バッファのアンダフロー，オーバフローを起こさないように滞りなくデータを転送することが求められる．

トランスコーダは，前章の議論をふまえ，量子化ステップをフレームごとに変更できる機構を備えている．量子化ステップはレート制御部から指定することができ，逆にレート制御部に対しては処理フレームのタイムスタンプ等，トランスコーダの内部情報を通知する．帯域推定およびレート制御アルゴリズムについては後述する．

### 3.1.3 クライアント

帯域推定，レート制御をサーバ側で行うため，クライアントとしては一般的なメディアプレーヤが利用できる．ネットワーク帯域の変動や一時的な切断の影響を軽減するため，プレーヤは受信バッファに一定時間のデータをバッファリングするものとする．適切なバッファリング量に関しては 4 章で検討を行う．

## 3.2 帯域推定およびレート制御アルゴリズム

レート制御部ではトランスコーダから受け取った情報をもとに現在のネットワーク帯域の推定を行い，それに適した動画のビットレートを決定し，トランスコーダの制御を行う．本節ではまず，既存研究においても用いられているフレームレートを用いる手法について概要およびその特徴について述べ，次に，本稿で提案するトランスコード遅延を用いる手法を述べる．その後これら 2 つの手法の比較を行い，提案ストリーミングシステムへの適用について検討する．

### 3.2.1 フレームレートを用いる手法

TCP ネットワークにおいては通信帯域の減少はパケット到達の遅延として現れる．これは，サーバ側から見れば送信バッファにデータが滞留し，データが用意されてから実際に送出されるまでの遅延が増大することになる．バッファ量は限られた値であるため，送信バッファでの遅延はそのままトランスコーダの処理速度の遅延となり，トランスコード処理のフレームレートが低下することになる．よって，動画自体のフレームレートを  $F$ ，トランスコードのフレームレートを  $f$  とすると，その時点でのネットワーク帯域  $W$  は，

$$W = \frac{f}{F} B \quad (1)$$

と推定できる．ここで  $B$  は動画コンテンツのビットレートである．このようにして得られた推定帯域を目標ビットレートとしてトランスコーダの量子化ステップを制御すれば，ネットワーク変動に応じた符号化レート制御が可能となる．この手法は既存研究<sup>5),8)</sup> においても採用されている．

しかし，この手法ではトランスコーダ処理速度の時間変化のみを用いるため，実際のネットワーク変動に対するレート制御の遅れが蓄積すると，クライアント側のバッファが枯渇する可能性がある．これを防ぐためにはトランスコーダで現在処理しているフレームのタイムスタンプ  $T_F$  と時刻  $t$  を比較し，トランスコード処理に遅延が発生していないかどうかを監視する必要がある．ここで，トランスコード遅延を  $d = t - T_F$  と定義する．実時間でトランスコードが行えている場合は  $d = 0$  となる．トランスコード遅延が  $d > 0$  となった場合は目標ビットレートを推定帯域より低い値に設定し，トランスコード遅延の回復を行う制御が必要となる．

### 3.2.2 トランスコード遅延を用いる手法

先に述べたフレームレートを用いる手法に対し，トランスコード遅延のみを使用し，直接帯域推定を行わずに符号化レート制御を行う手法が考えられる．以下，本稿で提案するレート制御アルゴリズムについて述べる．

本手法では，入力として現在時刻  $t$  およびトランスコーダが処理中のフレームのタイムスタンプ  $T_F$  を用いる．これらの値からトランスコード遅延  $d = t - T_F$  を求め，これに基づき目標ビットレート  $b$  を決定する．トランスコード遅延  $d$  に対する目標ビットレート  $b$  は以下により求められる．

- $d \leq 0$  の場合：

$$b = B_{\max}$$

ここで  $B_{\max}$  はコンテンツの最大ビットレート．

- $d > 0$  の場合：

$$b = \min(B_{\max}(1 - d/D_{\max}), B_{\min})$$

ここで  $B_{\min}$  はコンテンツの最小ビットレート， $D_{\max}$  は許容しうる  $d$  の最大遅延．

トランスコード遅延  $d$  と目標ビットレート  $b$  の関係を図 3 に示す．上記により決定された目標ビットレートをを用い，最終的に動画符号化の際の量子化ステップを制御する．

上記のようにレート制御を行った場合，ネットワーク帯域の変動に対する挙動は次のよう

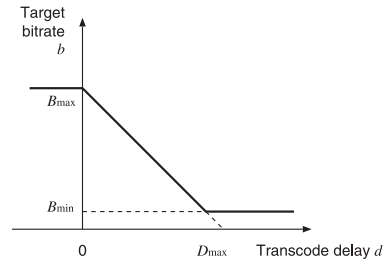


図 3 トランスコード遅延による符号化ビットレートの制御

Fig. 3 The target bitrate of video contents is controlled between  $B_{\max}$  and  $B_{\min}$  by transcode delay,  $d$ , within  $D_{\max}$ .

になる。ネットワーク帯域がコンテンツのビットレートに比べ十分に大きい場合はトランスコード遅延が発生しないため、コンテンツの最大ビットレートで配信を行う。ネットワーク帯域が減少するとトランスコード遅延  $d$  が増加するため、図 3 に従ってトランスコードの目標ビットレート  $b$  を低下させる。その結果、まだネットワーク帯域が不足する場合は引き続き  $d$  が増加するため、さらに  $b$  を下げる制御を行い、逆に下げすぎた場合には  $d$  が減少するため、より高画質にするために  $b$  の上昇を行う。これにより、その時点のネットワーク帯域に対し、トランスコード遅延が 0 以上  $D_{\max}$  以下のある値に収束し、安定することになる。なお  $D_{\max}$  は自由に設定可能であり、最大で  $D_{\max}$  の値分の遅延を許容することになるため、クライアント側でのバッファリング可能時間より小さく設定する必要がある。以降、本稿ではこのバッファリング可能時間を、キャッシュ時間と呼ぶことにする。トランスコード遅延  $d$  をクライアント側のキャッシュ時間より短く抑えることにより、動画を途切れさせることなく再生が可能となる。

### 3.2.3 フレームレート、トランスコード遅延を用いる各手法の比較検討

フレームレートを用いる手法は、観測されたフレームレートより推定帯域を算出し、それに基づき目標ビットレートを制御するという直感的に分かりやすい手法であり、帯域情報を取得する必要がある用途においては有効な手法である。しかし先に述べたように遅延が蓄積する可能性があり、クライアント側のバッファを枯渇させずに連続したストリーミングを行うためにはトランスコード遅延を監視し、目標ビットレートを制御する処理が別途必要になる。

これに対し、トランスコード遅延を用いる提案手法では帯域情報を直接得ることはできないが、トランスコード遅延を一定値以下に抑える制御をすることにより帯域変動に高速に

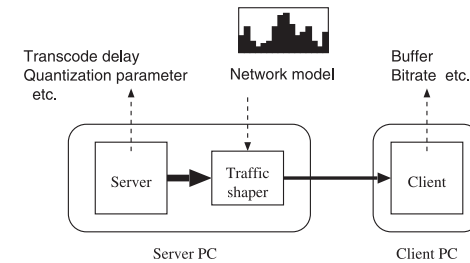


図 4 評価環境の概要

Fig. 4 An overview of the evaluation platform of the proposed streaming system.

追従し、クライアント側のバッファを枯渇させないようなレート制御を行うことができる。したがって、本研究で目指す途切れのないメディアストリーミングを実現するにあたり、トランスコード遅延を用いる本手法が適すると考えられる。

以上の理由により、本研究ではレート制御部にトランスコード遅延を用いる手法を採用し、ストリーミングシステムの構築および評価を行った。

## 4. 実験および評価

3章で提案した動的レート制御ストリーミングシステムの評価を行うため、サーバおよびクライアントをソフトウェア実装し、評価環境の構築を行った。

### 4.1 評価環境

提案手法の有用性を示すため、高速移動通信網において想定される様々なネットワーク帯域の変動をシミュレート可能な評価環境を構築した。概要を図 4 に示す。評価環境はサーバ、クライアントおよび高速移動通信網を模した仮想通信路から構成される。サーバ、クライアントはそれぞれ Linux 2.6.22 が動作する PC であり、これら 2 台が Ethernet により接続されている。TCP の輻輳制御アルゴリズムには CUBIC<sup>10)</sup> を利用した。サーバ PC の出力側には PC 間のトラフィック量を任意の値に制限できる Traffic shaper<sup>11)</sup> が設置されており、所望の帯域変動をシミュレートすることが可能となっている。サーバを構成するトランスコード、ストリーミングサーバはそれぞれオープンソースソフトウェアである ffmpeg、Live555 Media Server をもとに実装を行った。なおレート制御機構は ffmpeg 内部に付加的に実装した。クライアント側のメディアプレーヤについても同じくオープンソースソフトウェアの VLC Media Player をもとに実装した。トランスコードおよびメディアプレーヤ

からはビットレートやバッファ使用量等が観測可能である．

帯域変動パターンの作成には High Speed Downlink Packet Access (HSDPA) エミュレータを使用した．本エミュレータでは，端末の移動状態や受信状況により変化する SIR (Signal-to-Interference Ratio) をエミュレートし，それに応じ AMC (Adaptive Modulation and Coding scheme) 方式<sup>12)</sup>により変調方式および符号化率を動的に制御した結果の通信特性をエミュレートすることができる．AMC 方式による適応的な速度制御の結果，TCP 等の上位プロトコルでは急激な通信帯域の変動が見られることになる．上記のエミュレータを用い，受信状況の異なる複数の条件のもとで帯域変動を測定し，それらを組み合わせることにより帯域変動パターンを作成した．なお今回使用した HSDPA の通信速度は最大 7.2 Mbps であり，AMC の制御間隔は 2 ms である．

#### 4.2 実験

上記システムを用いて動画ストリーミングを行い，動的レート制御の効果について評価した．配信対象の動画は，フォーマットが MPEG-2 の Transport Stream (TS) 形式，画像サイズが CIF (352 × 288)，フレームレートが 29.97 fps，トランスコード前のビットレートが約 4,000 kbps のものを使用した．以下，高速移動通信網を想定した複数の帯域変動をシミュレートし，動的レート制御の評価を行った結果について述べる．

##### 4.2.1 帯域変動例 1

端末が時速 3 km での移動状態にあり，電波の受信状態が変動する場合の評価を行った．良好な通信品質より開始し，20 秒ごとに品質が低下し，その後再び回復するシナリオを用い，動的レート制御ストリーミングを行った．ネットワーク帯域の変動に対するトランスコード遅延，動画ビットレート，クライアントのバッファ使用量の測定結果を図 5～図 12 に示す．なおクライアントにおけるメディアプレーヤのキャッシュ時間は 10 秒間に設定した．

図 5，図 6，図 7，図 8 はトランスコードが許容する最大遅延  $D_{\max}$  を 1 秒としてレート制御を行った結果である．図より，ネットワーク帯域が約 4,000 kbps 以上の範囲ではトランスコード遅延は発生しておらず，動画のビットレートにも制限がかかっていないことが分かる．一方，帯域が減少するとそれにもとないトランスコード遅延が増大し，ビットレートが制限されていることが分かる．動画のビットレートは帯域の変動に即座に追従しており，ほぼネットワーク帯域と同じビットレートで安定して配信が行えている．また，図 8 より帯域減少時にもバッファが枯渇することなく再生を続けられていることが分かる．

図 9，図 10，図 11，図 12 は最大トランスコード遅延を  $D_{\max} = 5$  秒とした評価結果である．こちらも同様の結果となり，バッファが枯渇することなく再生が続けられていること

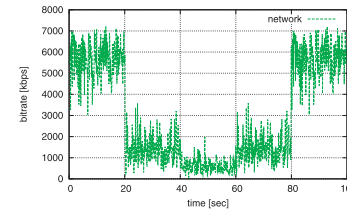


図 5 ネットワーク帯域 (変動例 1,  $D_{\max} = 1$  秒)  
Fig. 5 Network bandwidth (ex.1).

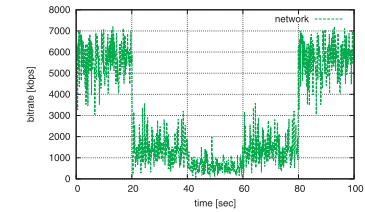


図 9 ネットワーク帯域 (変動例 1,  $D_{\max} = 5$  秒)  
Fig. 9 Network bandwidth (ex.1).

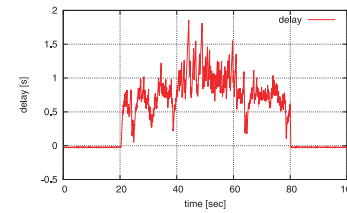


図 6 トランスコード遅延 (変動例 1,  $D_{\max} = 1$  秒)  
Fig. 6 Transcode delay (ex.1,  $D_{\max} = 1$  sec).

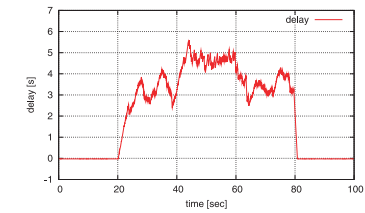


図 10 トランスコード遅延 (変動例 1,  $D_{\max} = 5$  秒)  
Fig. 10 Transcode delay (ex.1,  $D_{\max} = 5$  sec).

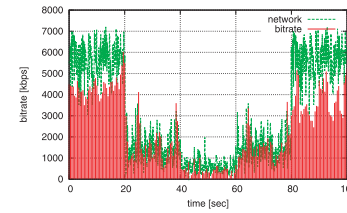


図 7 動画ビットレート (変動例 1,  $D_{\max} = 1$  秒)  
Fig. 7 Video bitrate (ex.1,  $D_{\max} = 1$  sec).

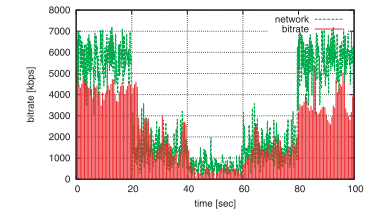


図 11 動画ビットレート (変動例 1,  $D_{\max} = 5$  秒)  
Fig. 11 Video bitrate (ex.1,  $D_{\max} = 5$  sec).

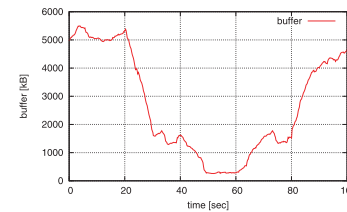


図 8 クライアントのバッファ使用量  
(変動例 1,  $D_{\max} = 1$  秒)  
Fig. 8 Client buffer usage  
(ex.1,  $D_{\max} = 1$  sec).

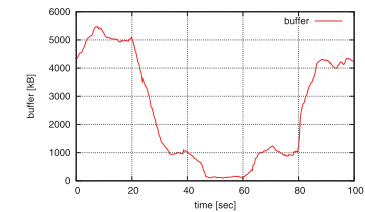


図 12 クライアントのバッファ使用量  
(変動例 1,  $D_{\max} = 5$  秒)  
Fig. 12 Client buffer usage  
(ex.1,  $D_{\max} = 5$  sec).

が分かる．なお  $D_{max}$  の違いによるストリーミングへの影響については後述する．

以上より，提案レート制御手法を用いることで，高速移動通信網における大幅な通信品質の変動に対しても動画を途切れさせることなく，ストリーミングを行えることが分かった．

#### 4.2.2 帯域変動例 2

次に，携帯端末等においてしばしば発生すると考えられるネットワーク切断を含む帯域変動例を用いた評価を行う．

端末は前項と同様に時速 3 km での移動状態にあり，良好な受信状態と切断状態が 10 秒ごとに繰り返されるシナリオを用いる．プレーヤのキャッシュ時間は 15 秒に設定した．評価結果を図 13～図 20 に示す．

前項と同様に図 13，図 14，図 15，図 16 は最大トランスコード遅延を  $D_{max} = 1$  秒，図 17，図 18，図 19，図 20 は  $D_{max} = 5$  秒としてレート制御を行った結果である．図 14 より，ネットワークが 10 秒間切断された場合，約 12 秒のトランスコード遅延が発生していることが分かる．これは今回実験に使用した TCP ネットワークの環境では，10 秒間のネットワーク切断が起こった際に，アプリケーションから見たセッションの切断時間が約 12 秒となることによる．ネットワークの非切断時はトランスコード遅延が 0 で最大ビットレートによる配信が行われ，ネットワークが切断されるとトランスコードが動作を停止する．その後セッションが回復すると，遅延を解消するために低ビットレートで配信が再開されていることが分かる．クライアントのパッファ量に関して枯渇することなく再生が行えており，提案レート制御手法によりネットワーク切断を含む帯域変動にも対応できることが示された．

#### 4.2.3 最大トランスコード遅延の検討

提案手法においては，図 3 に示したようにトランスコード遅延が最大値  $D_{max}$  を超えないようにビットレートの制御を行う．ここではこの値  $D_{max}$  の違いによるストリーミングへの影響について検討する．

まず帯域変動例 1 では， $D_{max} = 1$  秒の図 6 および  $D_{max} = 5$  秒の図 10 より， $D_{max}$  の値が大きい方がトランスコード遅延のばらつきが抑えられていることが分かる．本手法ではトランスコード遅延の値を直接動画のビットレート制御に用いるため，ばらつきが小さい方が画質の変動の少なく，主観画質が向上すると考えられる．なお，図 7，図 11 に示したビットレートの測定結果においては  $D_{max}$  の違いによる差異があまり見受けられないが，これはビットレートの測定間隔が変動周期より大きかったためと考えられる．

帯域変動例 2 では 12 秒という長時間のネットワーク切断により，図 15，図 19 に示したように  $D_{max}$  を超えるトランスコード遅延が発生している．このような場合，ネットワーク切

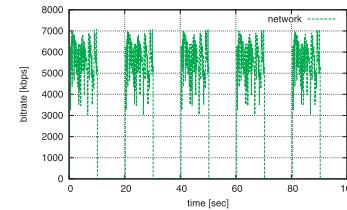


図 13 ネットワーク帯域 (変動例 2,  $D_{max} = 1$  秒)  
Fig. 13 Network bandwidth (ex.2).

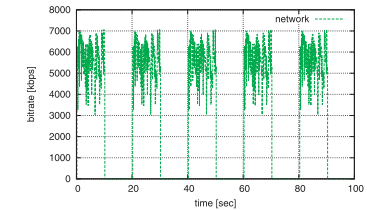


図 17 ネットワーク帯域 (変動例 2,  $D_{max} = 5$  秒)  
Fig. 17 Network bandwidth (ex.2).

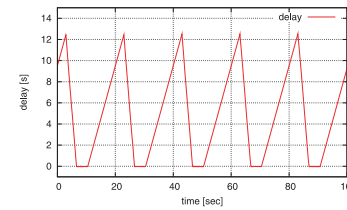


図 14 トランスコード遅延 (変動例 2,  $D_{max} = 1$  秒)  
Fig. 14 Transcode delay (ex.2,  $D_{max} = 1$  sec).

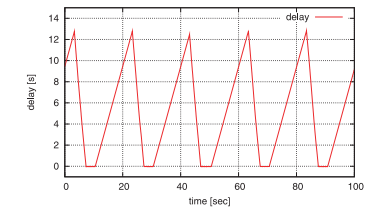


図 18 トランスコード遅延 (変動例 2,  $D_{max} = 5$  秒)  
Fig. 18 Transcode delay (ex.2,  $D_{max} = 5$  sec).

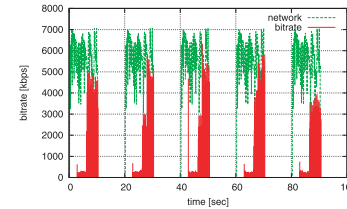


図 15 動画ビットレート (変動例 2,  $D_{max} = 1$  秒)  
Fig. 15 Video bitrate (ex.2,  $D_{max} = 1$  sec).

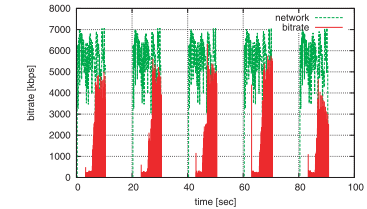


図 19 動画ビットレート (変動例 2,  $D_{max} = 5$  秒)  
Fig. 19 Video bitrate (ex.2,  $D_{max} = 5$  sec).

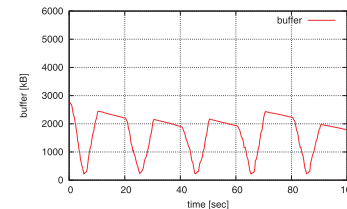


図 16 クライアントのパッファ使用量  
(変動例 2,  $D_{max} = 1$  秒)  
Fig. 16 Client buffer usage  
(ex.2,  $D_{max} = 1$  sec).

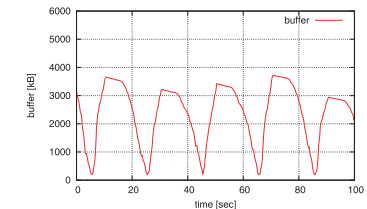


図 20 クライアントのパッファ使用量  
(変動例 2,  $D_{max} = 5$  秒)  
Fig. 20 Client buffer usage  
(ex.2,  $D_{max} = 5$  sec).



断からの復帰時、つまりトランスコード遅延  $d$  が減少している状況を考えると、 $D_{\max}$  の値が大きい方が  $d > D_{\max}$  となる時間が短くなる。すなわち、 $D_{\max}$  が大きい方が迅速にビットレートを上昇させる制御を行うことになり、結果として平均ビットレートを高く保つことができ、動画の画質を向上させることが可能となる。実験結果においても、 $D_{\max} = 1$  秒の図 16 に比べ、 $D_{\max} = 5$  秒の図 20 の平均ビットレートが高くなっていることが分かる。

以上の結果より、 $D_{\max}$  の値は大きい方がビットレートの変動を抑えられ、かつクライアントのバッファ使用量を増加させることができ、画質向上に効果的であることが分かった。ただし、次項で述べるようにクライアントのキャッシュ時間は  $D_{\max}$  秒以上必要であるため、システムの性能に応じ実現可能な範囲内に設定する必要がある。

#### 4.2.4 クライアントバッファ量の検討

ネットワークにおいて帯域変動や一時的な切断が生じた場合、クライアント側のバッファが枯渇し動画の再生が途切れる可能性がある。このような動画の中断を回避するためにはネットワークの特性、性質に応じ、クライアント側で十分なバッファリングを行う必要がある。以下、ネットワーク帯域の変動や一時的な切断を許容するためにクライアント側に必要なバッファ量について検討する。

まず、帯域変動例 1 のようにネットワーク切断のない場合を考える。図 6、図 10 のトランスコード遅延の遷移より、遅延時間はおよそ  $D_{\max} + 1$  秒以内に収まっていることが分かる。よって、ネットワーク切断を想定しない場合は、プレーヤのキャッシュ時間は  $D_{\max} + 1$  秒以上に設定することによりバッファの枯渇を回避し、連続したストリーミングが可能になるといえる。動画の最大ビットレートが  $B_{\max}$  のとき、クライアントに必要なバッファ量は  $B_{\max}(D_{\max} + 1)$  となる。

一方、帯域変動例 2 のようにネットワーク切断をとまなう場合は、 $D_{\max} + 1$  秒以上、かつ切断時間  $t_{\text{int}}$  以上のキャッシュ時間を設定することでバッファの枯渇を回避できる。つまり必要なバッファ量は、 $\max(B_{\max}(D_{\max} + 1), B_{\max}t_{\text{int}})$  となる。ここで、本手法では TCP ネットワーク上でストリーミングを行うため、物理層における切断時間を  $t_{\text{phy}}$  とすると、TCP 層での通信再開に要する時間  $t_{\text{TCP}}$  を加えた  $t_{\text{int}} = t_{\text{phy}} + t_{\text{TCP}}$  を切断時間として考える必要がある。今回評価に使用したネットワーク環境では、 $t_{\text{phy}}$  を変化させたときの  $t_{\text{int}}$  を測定したところ、表 1 のとおりであった。 $t_{\text{int}}$  が離散的な値となっているのは今回利用した Linux における TCP の実装によるもので、6 秒間隔で接続の再試行が行われるためであると考えられる。実際の携帯端末に適用する際には、表 1 のようなネットワークに関するデータを使用し、システムとして許容する切断時間  $t_{\text{phy}}$  をもとに、それに適したバッ

表 1 評価環境におけるネットワーク切断時間

Table 1 Network interruption time observed under the evaluation environment.

$t_{\text{phy}}$	$t_{\text{int}}$
0 sec	0 sec
0 ~ 6 sec	6 sec
6 ~ 12 sec	12 sec
12 ~ 18 sec	18 sec

ファ量を決定することになる。

なお、上記のようにバッファ量を設定した場合でも、端末が圏外に長時間とどまる等、想定外のネットワーク切断が発生した場合はクライアント側のバッファが枯渇し、ストリーミング再生が停止することになる。この場合、ストリーミングの中断は不可避であり、クライアント側はネットワークが回復し次第、届いたストリーミングデータから順に再生を再開する。

## 5. 結 論

本研究では、ネットワーク経由での動画配信において、ビットレートを動的に制御することにより画像品質を向上させることを目的とし、動的レート制御ストリーミングシステムの提案を行った。特に固定ネットワークと比べ帯域変動が激しい高速移動通信網においても、動画を途切れさせることなく高品質なストリーミングを行うため、トランスコード処理の遅延時間を用いて直接ビットレートを決定する手法を採用した。本研究では提案手法の評価を行うため、動的レート制御システムを実装し、高速移動通信網において想定される複数の帯域変動パターンを用いた評価を行った。その結果、急激な帯域変動に対しても迅速にレート制御を行い、クライアント側でのバッファを枯渇させることなく連続したストリーミングが行えることを示した。

謝辞 本研究の一部は有限責任中間法人 PUC (P2P Universal Computing Consortium)、および文部科学省 G-COE プログラム「光・電子理工学の教育研究拠点形成」の支援を受けて実施された。

## 参 考 文 献

- 1) Tsutsui, H., Yu, J., Izumi, T., Ochi, H., Nakamura, Y., Komura, T., Uchida, Y. and Ishikawa, N.: Implementation of AV Streaming System Using Peer-to-Peer Com-

- munication, *Proc. IEEE Consumer Communications and Networking Conference*, pp.778–782 (2007).
- 2) Bolot, J. and Turletti, T.: A Rate Control Mechanism for Packet Video in the Internet, *Proc. Conference on Computer Communications*, pp.1216–1223 (1994).
  - 3) Puri, R., Lee, K., Ramchandran, K. and Bharghavan, V.: An Integrated Source Transcoding and Congestion Control Paradigm for Video Streaming in the Internet, *IEEE Trans. Multimedia*, Vol.3, No.1, pp.18–32 (2001).
  - 4) de Cuetos, P. and Ross, K.: Adaptive Rate Control for Streaming Stored Fine-Grained Scalable Video, *Proc. 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp.3–12 (2002).
  - 5) Takaoka, T., Nagayoshi, I., Hanamura, T. and Tominaga, H.: Network Adaptive Rate Control for Transcoder, *Proc. 13th European Signal Processing Conference* (2005).
  - 6) Lei, Z. and Georganas, N.: Rate Adaptation Transcoding for Video Streaming over Wireless Channels, *Proc. IEEE International Conference on Multimedia and Expo*, Vol.2, pp.433–436 (2003).
  - 7) 酒澤茂之, 宮地悟史, 滝嶋康弘, 和田正裕: TCP ビデオストリーミングの動的レート制御方式の検討, 電子情報通信学会技術研究報告, IE, 画像工学, Vol.102, No.469, pp.19–24 (2002).
  - 8) 中平航太, 永吉 功, 花村 剛, 富永英義: TCP の輻輳制御によるトランスコーダへの影響を利用した動的ビットレートスケールリング手法, 電子情報通信学会技術研究報告, IE, 画像工学, Vol.102, No.632, pp.141–146 (2003).
  - 9) Postel, J.: RFC 793: Transmission Control Protocol, DARPA Internet Program Protocol Specification (1981).
  - 10) Rhee, I. and Xu, L.: CUBIC: A New TCP-Friendly High-Speed TCP Variant, *Proc. PFLDnet* (2005).
  - 11) LWN.net: The Linux Traffic Shaper. <http://lwn.net/1998/1119/shaper.html>
  - 12) 3rd Generation Partnership Project: Technical Specification Group Radio Access Network; Physical Layer Aspects of UTRA High Speed Downlink Packet Access, Technical Report 25.848 (2001).

(平成 20 年 12 月 16 日受付)

(平成 21 年 7 月 2 日採録)

## 推 薦 文

本論文は, 移動通信端末のためのストリーミング配信において通信品質の変化に応じ動画ビットレートの動的制御を行う方式を提唱している. 本方式では, サーバ側でトランス

コード処理の遅延を検出することでネットワークの変動を感知している. また, 高速移動通信網において発生する複数の帯域変動パターンに対して評価を行っている. 高い実用性は論文誌の推薦論文としてふさわしい.

(マルチメディア, 分散, 協調とモバイル (DICOMO2008) シンポジウム  
プログラム委員長 串間和彦)



廣本 正之 (学生会員)

2006 年京都大学工学部電気電子工学科卒業. 2007 年同大学院情報学研究科通信情報システム専攻修士課程修了. 2007 年同博士後期課程進学. 2008 年より日本学術振興会特別研究員 (DC1), 現在に至る. リコンフィギャラブルシステム, 組み向け画像認識およびその実装に関する研究に従事. 電子情報通信学会, IEEE 各学生会員.



筒井 弘

2000 年京都大学工学部電気電子工学科卒業. 2002 年同大学院情報学研究科通信情報システム専攻修了. 2005 年同博士後期課程修了. 同年同大学特任助手. 2007 年より大阪大学大学院情報科学研究科特任助教. 組み向け画像処理システムおよびその実装に関する研究に従事. 博士 (情報学). IEEE, ACM, 電子情報通信学会, 電気学会各会員.



越智 裕之 (正会員)

1989 年京都大学工学部情報工学科卒業. 1991 年同大学院工学研究科修士課程情報工学専攻修了. 1994 年同博士後期課程修了. 同年広島市立大学情報科学部情報工学科助教授. 2004 年京都大学大学院情報学研究科通信情報システム専攻助教授, 現在に至る. 再構成アーキテクチャ, 算術演算回路, 低消費電力設計, 画像符号化等の研究に従事. 博士 (工学). 電子情報通信学会, IEEE, ACM 各会員.



小佐野智之

2003年千葉大学工学部電子機械工学科卒業。2005年同大学院自然科学研究科電子情報システム専攻修士課程修了。同年(株)NTTドコモ入社。モバイルインターネットプロトコルおよびアプリケーションの研究開発に従事。



石川 憲洋 (正会員)

1978年京都大学工学部情報工学科卒業。1980年同大学院工学研究科情報工学専攻修士課程修了。同年日本電信電話公社(現,NTT)入社。現在,NTTドコモサービス&ソリューション開発部に所属。モバイルインターネット,ユビキタスコンピューティング等の研究開発に従事。博士(情報学)。電子情報通信学会シニア会員,IEEE会員。



中村 行宏 (正会員)

1967年京都大学工学部数理工学科卒業。1969年同大学院工学研究科数理工学専攻修士課程修了。同年日本電信電話公社入社。NTT情報通信研究所高速通信処理研究部長等を経て,1996年京都大学教授。2007年同定年退職。同年立命館大学教授,(財)京都高度技術研究所所長,副理事長,現在に至る。並列処理アーキテクチャとその設計方式,マルチメディアシステム,デジタル社会基盤の研究・開発に従事。博士(工学)。大河内記念技術賞,科学技術庁長官賞,電子情報通信学会業績賞,近畿総合通信局長表彰等各受賞。電子情報通信学会,IEEE各会員。IEEE Fellow,IEEE関西支部長,NPOパルテノン研究会副理事長等。