

## [招待講演] サービス品質向上のためのネットワーク状態推定・予測技術

里田 浩三<sup>†</sup> 大芝 崇<sup>†</sup> 吉田 裕志<sup>†,††</sup> 中島 一彰<sup>†</sup>

<sup>†</sup> NEC クラウドシステム研究所 〒211-8666 神奈川県川崎市中原区下沼部 1753

<sup>††</sup> 東京工業大学 情報理工学研究所 〒152-8550 東京都目黒区大岡山 2-12-1

E-mail: <sup>†</sup> {k-satoda@cb, oshiba@cp, h-yoshida@jh, nakajima@ah}.jp.nec.com <sup>††</sup> yoshida.h.aj@m.titech.ac.jp

**あらまし** 端末、ネットワークの高速化、高性能化に伴い、様々なサービスがインターネットやモバイルネットワーク経由で利用されるようになってきている。しかし、これらベストエフォートでは、クロストラヒックや電波状態によって利用できるネットワーク帯域が大きく変動し、クラウドサービス利用者の体感品質を低下させる原因となっている。本稿では、ベストエフォートネットワークにおけるクラウドサービスの利用品質を向上するため、ネットワークの現在の状態をいち早く推定するもしくは未来の状態を予測するネットワーク状態推定・予測技術について紹介する。また、ネットワーク状態推定・予測を利用したアプリケーション例を紹介する。

**キーワード** クラウドサービス, 可用帯域, スループット, 推定・予測

## Network Traffic Estimation and Prediction Technologies for Improving User-perceived Quality

Kozo SATODA<sup>†</sup>, Takashi OSHIBA<sup>†</sup>, Hiroshi YOSHIDA<sup>†,††</sup>, and Kazuaki NAKAJIMA<sup>†</sup>

<sup>†</sup> Cloud System Research Laboratories, NEC Corporation

1753 Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211-8666 Japan

<sup>††</sup> Graduate School of Information Science and Engineering, Tokyo Institute of Technology

2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550 Japan

E-mail: <sup>†</sup> {k-satoda@cb, oshiba@cp, h-yoshida@jh, nakajima@ah}.jp.nec.com <sup>††</sup> yoshida.h.aj@m.titech.ac.jp

**Abstract** Cloud services via the Internet and mobile networks have been popular thanks to faster networks. However, best effort networks such as the Internet and mobile networks are fluctuated widely due to cross-traffics and signal interferences and it causes deteriorating the user-perceived quality. This paper introduces network traffic estimation technologies, which realizes higher user-perceived quality of cloud services over best effort networks. This paper also describes application examples of the network traffic estimation technologies.

**Key words** cloud services, available bandwidth, throughput, estimation, prediction

### 1. はじめに

端末、ネットワークの高速化、高性能化に伴い、様々なサービスがインターネットやモバイルネットワーク経由で利用されるようになってきている。しかし、これらベストエフォートでは、クロストラヒックや電波状態によって利用できるネットワーク帯域が大きく変動し、クラウドサービス利用者の体感品質を低下させる原因となっている。

クラウドサービス事業者にとって、このようなベストエフォートネットワークを利用しても、高品質なサービスを提供することが、他のサービスに対する差異化となりうる。また、通信キャリ

アにとっても、モバイルネットワークを利用しても高品質な通信サービスを提供することが、他の事業者に対する差異化となる。

本稿では、ベストエフォートネットワークにおけるクラウドサービスの利用品質を向上するため、ネットワークの現在の状態をいち早く推定するもしくは未来の状態を予測するネットワーク状態推定・予測技術について紹介する。また、ネットワーク状態推定・予測を利用したアプリケーション例を紹介する。

### 2. ネットワーク状態推定・予測

本稿で紹介するネットワーク状態推定・予測およびその応用例は、クラウドサービス事業者がネットワーク状態をモニタリング

し、モニタリング結果に応じて、通信を最適化することで、ユーザにとって高品質なサービスを提供することを想定している。

ユーザが高い体感品質を感じるためには、ネットワーク機器で QoS を確保するだけでなく、ユーザに近いレベルでの品質確保が必要である。Tasaka [1] らは、これを「ネットワークレベル QoS」に対して「ユーザレベル QoS」と呼んだ。しかし、ユーザレベル QoS はユーザの主観品質と大きく関係があるため、クラウド事業者のシステムでユーザレベル QoS を確保することは困難である。我々はクラウドサービスを提供するサーバおよびユーザの端末だけで通信品質をモニタリングし、通信を制御することでサービス品質向上をめざす。Tasaka らの分類では、ネットワークレベルより高く、ユーザレベルより低い、「エンド・ツー・エンドレベル」および「アプリケーションレベル」でのサービス品質を想定する。ここでは、通信の端点のみ、またはクラウドサービスのアプリケーションで入手できる情報のみを利用して、通信品質をモニタリングすることを、「エンド・ツー・エンドレベル」および「アプリケーションレベル」のネットワーク推定・予測と呼ぶ。

通常、クラウドサービス事業者は、ユーザ端末へ直接つながるネットワークを保有せず、通信キャリアのネットワークを利用しているだけであるため、サービス品質向上のためには、ネットワークに手を入れて品質を向上するアプローチを採用することはできない。エンド・ツー・エンドレベルおよびアプリケーションレベルの方式は、通信の端点、クラウドサービスのアプリケーションで入手できる情報のみを利用しており、クラウドサービス事業者が導入しやすい。通信キャリアにとってもメリットがある。ネットワークレベル QoS は、通信機器にモニタリングのための仕掛けを組み込む必要がある。さらに通信制御のためには通信経路や優先度を変更可能な通信装置を用意しなければならない。このようにネットワークレベルのアプローチでは多大な設備投資が必要になるが、アプリケーションレベル、エンド・ツー・エンドレベルのアプローチは通信の端点でのみモニタリング・制御の実施が可能である。通信キャリア網内にサーバ・端末間の通信を一旦終端するサーバを用意するだけでモニタリング・制御が可能となるため、通信キャリアでも容易に機能配備できる。

本推定・予測方式は、今実施中もしくは実施しようとしている通信を最適化することを想定している。長時間の通信履歴やトラフィック履歴から通信特性やネットワーク特性を学習するのでは実施中の通信の最適化には間に合わない。このため、リアルタイムに推定・予測することで、実施中の通信を素早く最適化するアプローチを採用する。

以上、我々のネットワーク状態推定・予測の技術アプローチをまとめると、次のようになる

- アプリケーションレベルおよびエンド・ツー・エンドレベルで推定・予測を実施する。
- リアルタイムに推定・予測を実施する。

### 3. 短時間可用帯域推定

ベストエフォートネットワークを介したクラウドサービスで

は、サービスが利用できる帯域 (可用帯域) が時々刻々と変動し、十分な帯域が得られずにユーザの体感品質が劣化することがある。スマートフォンには接続しているネットワーク (LTE, 3G, 無線 LAN など) や電波強度は表示されているが、実際に通信できる帯域は表示されていないため、ユーザは表示されている状態よりもサービス品質が悪いと判断してしまうことがある。

クラウドサービス事業者は、サーバと端末間で利用できる帯域を測定し、可用帯域に合わせてデータ量を調整して配信することで、体感品質劣化を防ぐことができる。一般的な帯域計測方法は、サーバと端末間でプローブパケットを送信し、プローブパケットの挙動により、帯域を測定するが、既存の手法では、

- プローブパケットのデータ量が大きく (数 MB 以上)、通信負荷がかかる。
- 測定に数秒から数十秒の時間がかかり、すぐに結果が得られない。

という課題があった [2] [3]。これに対し、86kB のプローブパケットの転送で、0.2 秒で可用帯域を推定できる PathQuick を提案している [4] [5] [6]。

本章では、短時間可用帯域推定技術 PathQuick とその応用例を紹介する。

#### 3.1 短時間可用帯域推定方式

短時間可用帯域推定技術 PathQuick の可用帯域推定方式を図 1 に示す。PathQuick では送信端末が、連続したパケットで構成されるパケットトレインを送信し、受信端末で到着するパケットの到着間隔を監視することで可用帯域を推定する。送信端末が送信するパケットトレインは、等間隔にパケットを配置し、それぞれのパケットサイズが線形に増加するように構成されている。すなわち、送信端末からこのパケットトレインを送出すると、送信レートが徐々に高くなるようになっている。受信端末ではパケットトレインの各パケットの到着間隔を監視する。送信端末の送信間隔と、受信端末での到着間隔が同じ場合は、送信レートが可用帯域に達していない。しかし、徐々に大きなパケットを送信していくと、受信端末でのパケットの到着間隔が伸びていく。パケットの到着間隔が伸びる点を検出し、その点のパケットサイズから送信レートを計算することで、可用帯域を推定する。

クラウドサービスでは、クラウドサービスのサーバを送信端末、端末を受信端末とすればクラウドサービスの下りの可用帯域が、端末を送信端末、サーバを受信端末とすれば上りの可用帯域が推定できる。

既存方式は、パケットサイズが一定のパケットトレインを何度も測定したり、大きなデータサイズの通信を行い、通信にかかった時間から可用帯域を推定しているため、通信データ量が大きく、かつ推定に時間がかかっていた。PathQuick では、送信するパケットを線形に増加させることで、少ないデータ量のパケットトレインを一回送信するだけで推定が可能となる。

#### 3.2 短時間可用帯域推定の応用例

短時間可用帯域推定は、少ないデータ量で短時間にエンド・ツー・エンドの可用帯域が推定できるという特徴を活かして、様々な応用例が考えられる。以下で、2 例を紹介する (図 2)。

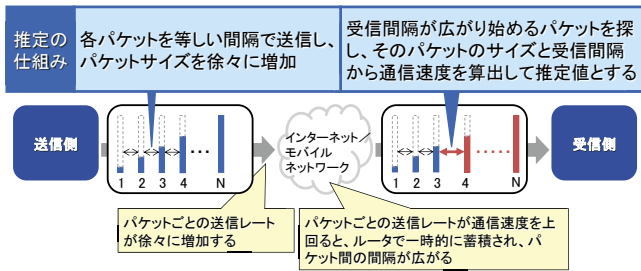


図1 短時間可用帯域推定方式

### 3.2.1 通信速度モニタリング

短時間可用帯域推定技術は、多数の端末の通信速度のモニタリングに利用できる。大量のデータを通信して通信速度を測定するスピードテストサイトが、インターネット上に多数存在するが、このような方式では多数の端末が利用可能帯域（通信速度）を測定すると、大きなネットワーク負荷をかけてしまう。また、データ通信量に制限がある料金プランに加入しているユーザは、大きなデータを使ったモニタリングを何度も行うと、利用制限に引っかかる可能性が出てくる。

短時間可用帯域推定技術は少ないデータ量で可用帯域を推定できるため、多数の端末の可用帯域を高い頻度で測定することができる。通信速度を測定するサーバをサービス事業者や通信キャリアの網に配備し、端末とサーバとの可用帯域をモニタリングすることで、サービス品質の推定、設備計画などに利用可能である。

### 3.2.2 通信経路選択

現在利用されているスマートフォンはモバイル網（3G/LTE）と無線LANを切り替えて利用することができる。通信キャリアはモバイル網の負荷を軽減するため、利用者が多い場所に無線LANホットスポットを用意し、通信トラフィックの一部を無線LAN経由で流すように推奨していることも多い（トラフィックオフロード）。しかし、無線LANをホットスポットが混雑しており、無線LANに接続したがために通信速度が下がり、ユーザが不満を感じるケースもある。

これを防ぐため、モバイル網と無線LANの両方が利用可能である場合、通信速度が速い方のネットワークに接続することで、快適なデータ通信を実現することができる。実際に通信速度を測定することで、電波強度だけではわからない通信速度を判断基準にして切り替えることができる。PathQuickでは短時間で通信速度を測定することができるため、モバイル網経由の通信速度と無線LAN経由の通信速度を素早く測定して、ユーザに不便を感じさせることなく切り替えることが可能となる。

## 4. 通信スループット予測

短時間可用帯域推定により、現在の可用帯域を0.2秒以内で知ることができる。しかし、可用帯域が瞬時にわかるとはいえ、推定した時点では過去になっている。一方、モバイルネットワークでは電波状態の大きな変動により、エンドツーエンドの通信スループットが不安定に変動することが確認されている[7]。このような大きく変動する環境では、現在の可用帯域を素早く推定

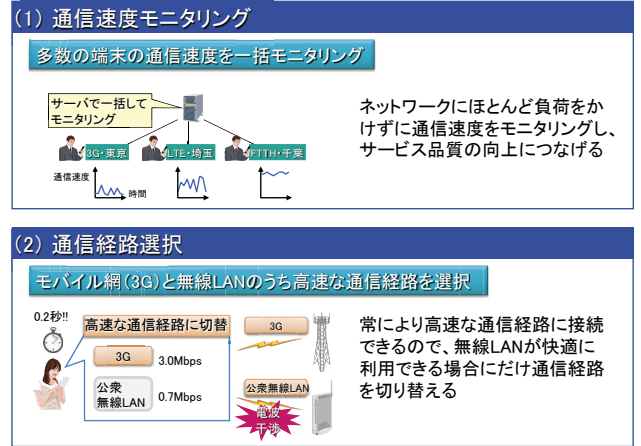


図2 短時間可用帯域推定方式の応用例

するだけでなく、未来の通信スループットを予測し、予測結果に合わせて通信制御することで高品質なクラウドサービスの実現を目指す。

本章では、アプリケーションレベルで未来の通信スループットを予測する方式と、そのアプリケーションとして映像配信サービスの例を紹介する。

#### a) 通信スループット予測方式

本章で紹介する通信スループット予測方式は、クラウドサービス事業者が通信スループットを予測し、通信制御を行うことを想定し、アプリケーションレベルで得られる通信データ量の履歴のみを利用する。また、ベストエフォート網では、通信スループットの変動が激しく、未来の通信スループットをピンポイントで確定的に予測することは困難であると考え、本方式では、未来の通信スループットを確率変数として扱い、確率的な広がり（確率的拡散）を予測する[8]。

図3に、通信スループット予測の予測方法を示す。図3(3)が通信スループットの確率的拡散予測を表している。すなわち、過去の通信スループットの履歴をもとに、未来の不確かな通信スループットを予測する。この時、決定的に未来の通信スループットを求めるのではなく、未来の通信スループットがどの範囲に分布しているかを確率的に求める。

次に、通信スループットの予測方法を説明する。通信スループット予測には、図3のように、通信スループットの定常性解析、予測モデル構築、確率的拡散予測の順で予測を実施する。以下で、それぞれについて概要を述べる。

#### b) 通信スループットの定常性解析

通信スループットの変動には、一定値周りを安定して変動する定常状態と、不安定に乱高下する非定常状態とが混在している。通信スループット予測では、まず通信状態が定常状態にあるのか、非定常状態にあるのかを判別する。

状態を判別するために、非定常状態を表すモデル( $M_n$ )と定常状態を表すモデル( $M_s$ )を定義し、統計的仮説検定を用いる。非定常モデルは特性方程式に単位根を持つことから、統計的仮説検定のひとつである単位根検定を用いる。非定常状態を帰無

### ① 通信スループットの定常性解析



### ② 予測モデル構築



### ③ 確率的拡散予測

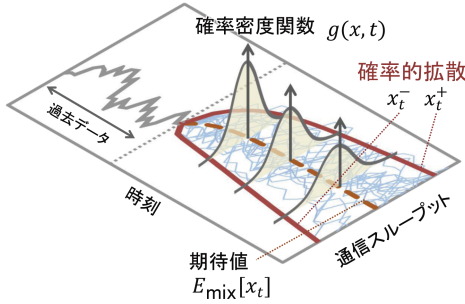


図 3 通信スループットの予測モデルの構築と確率的拡散予測

仮説，定常状態を対立仮説とおき，単位根判定にて定常状態か非定常状態かを判別する。

#### c) 予測モデル構築

通信スループットの定常性の判別結果にもとづき，通信スループットの予測モデルを構築する．判別された状態におけるモデル ( $M_n$  もしくは  $M_s$ ) を予測モデルとして採用するのが最も簡単である．しかし，変動状態も時々刻々と変動しているため，ここでは予測モデルも定常モデルと非定常モデルが確率的に混合されていると考える．混合比は過去の判別結果の履歴に基づいて計算する．時刻  $t$  [秒] における通信スループットを  $x(t)$  [bps] とし，非定常モデル，定常モデルの確率密度関数をそれぞれ  $f_n(x, t)$ ， $f_s(x, t)$  としたとき，予測モデルの確率密度関数  $g(x, t)$  は，

$$g(x, t) = r_t f_n(x, t) + (1 - r_t) f_s(x, t) \quad (1)$$

で得られる．ここで， $r_t$  は定常状態と非定常状態の混合比である．

#### d) 確率的拡散予測

構築した予測モデルを用いて，下記で定義した通信スループットの確率的拡散を予測する．

$$x_t^\pm = E_{\text{mix}}[x_t] \pm \alpha \sqrt{V_{\text{mix}}[x_t]} \quad (2)$$

ここで， $E_{\text{mix}}[x_t]$ ， $V_{\text{mix}}[x_t]$  はそれぞれ予測モデル  $g(x, t)$  の期待値と分散である．確率的拡散は期待値から標準偏差の定数倍 ( $\alpha$  倍) だけ加減した値であり，時間経過に伴う通信スループットの広がりを示す．図 3(3) に確率密度関数  $g(x, t)$ ，期待値  $E_{\text{mix}}[x_t]$ ，確率的拡散  $x_t^\pm$  の関係を示した．

### 4.1 通信スループット予測の応用例

インターネットやモバイルネットワークを介した映像配信サー

ビスが増加している．インターネット上の映像配信の多くが，映像配信プロトコルとして従来から用いられてきた UDP ではなく，HTTP/TCP が用いている．HTTP が用いられる主な理由はシステム構築が安価であることやファイアウォールが用意に通過できることが挙げられる．これまで述べてきたように，インターネットやモバイルネットワークは通信スループットが大きく変動するため，配信する映像のビットレートに対して通信スループットが不十分で，映像再生の停止が頻発し，ユーザの体感品質が悪化するという課題がある．

また，画像やテキストに比べて映像データはデータ量が多いため，映像配信サービスのトラフィックが通信キャリアのトラフィックの多くを占めており，いかに効率の良く映像トラフィックを処理するかという課題も生じている．HTTP/TCP を利用したインターネット映像配信サービスでは，映像ファイルをダウンロードし，クライアントの再生バッファに蓄積しながら，一定量のデータが蓄積された時点で再生を開始するプログレッシブダウンロード方式を用いている．プログレッシブダウンロードによる映像配信では，クライアントは通信スループットの許す限り映像データを取得し続けるため，途中で視聴をやめた場合に未再生のダウンロード済みデータが無駄になってしまう．代表的なインターネット映像配信サービスである YouTube では 60% の映像配信セッションは全体の 20% 以下しか視聴されておらず，25% から 39% のダウンロード済みデータが視聴されずに破棄されていると報告されている [9]．特に通信キャリアにとって，利用されないデータがダウンロードされることは通信資源の無駄になり，大きな課題になっている．

本節では，通信スループット予測の応用例として，予測に基づいて映像配信を制御することで，上記課題を解決する適応映像配信方式とビデオペーシング方式を紹介する．

#### 4.1.1 適応映像配信

上記で述べてきたように，インターネットやモバイルネットワークは通信スループットが大きく変動するため，配信する映像のビットレートに対して通信スループットが不十分で，映像再生の停止が頻発することがある．これに対し，通信スループットに応じて映像ビットレートを制御するような配信システムが登場してきている [10] [11] [12]．いずれも，一つの映像コンテンツを短い時間間隔で分割し，分割したそれぞれのセグメント (チャック) は複数のビットレートを持つように構成されている．各セグメントを配信する際に，通信スループットに応じて複数のビットレートから配信するビットレートを決定して配信することで，変動する通信スループットに適応できるようになっている．このような映像配信システムを「適応映像配信」と呼び，国際標準として規定された映像フォーマット，プロトコルも存在する [13]．これら適応映像配信システムは通信スループットが変動する環境において，ある程度有効性が示されているが，大きな通信スループット変動への適応性が低く，高い体感品質を得るためにはまだ改善が必要であるという報告もある [14]．

通信スループット予測の応用例として，予測に基づく映像ビットレートを動的に制御する方式を紹介する [15] [16]．予測した通



信スループットに基づくことで、通信スループットが大きく変動する環境でも、既存の制御方式より映像の途切れが発生せず、高品質な映像視聴が可能である。本適応映像配信の制御目的は、再生バッファ枯渇による映像の途切れが発生しないという条件の下で、できる限り高品質（高ビットレート）の映像視聴を実現する。

本方式の映像ストリーミングシステムは、サーバから HTTP/TCP を利用して映像コンテンツをクライアントに配信する。映像ストリーミングシステムの構成例を図 4 に示す。映像コンテンツは一つのコンテンツが  $m$  個のビットレートによって符号化されている。ビットレートの集合を  $\mathfrak{B} = \{b_1, b_2, \dots, b_m\}$  とする。また、コンテンツは一定時間 ( $l$  秒) ごとに分割され、それぞれのセグメントごとに切り替え可能のように構成されている。

次にビットレート集合  $\mathfrak{B}$  からビットレートを選択する方法を考える。図 4 に示した映像配信システムに対して、映像配信を支配する方程式は次のようになる [17]。

$$\frac{dT_p}{dt} = \frac{x}{r} - p \quad (3)$$

$T_p$  はクライアントの再生バッファ量 (単位: 秒),  $x$  は通信スループット (単位: bps),  $r$  は映像ビットレート (単位: bps) である。  $p$  はクライアントでの再生速度であり、早送りや巻戻しをせず、等速で再生している場合  $p = 1$ 、二倍速で再生している場合  $p = 2$  となる。  $T_p = 0$  となると再生バッファが空になり、映像再生が停止する。

未来の通信スループットが悪化に推移した場合でも  $T_p = 0$  とならないように制御する。すなわち、式 (2) で示した未来の通信スループット変動  $x^\pm(t)$  の下側  $x^-(t)$  で推移したとして、ある時刻  $t$  から  $T$  秒後の再生バッファ量  $T_p(T)$  秒を求める。求めた  $T_p(T)$  を目標値  $T_r$  に近づけるように映像ビットレート  $r$  を制御し、ビットレート集合  $\mathfrak{B}$  から  $r$  を超えないビットレートを選択する。

通信スループット予測を用いた適応映像配信方式と予測を用いない適応映像配信方式の性能比較を行った。予測を用いない方式は、 $x^-(t)$  ではなく、現時点での TCP スループット  $x_0$  を用いて制御を実施する。利用した蓄積映像は、300 秒のコンテンツであり、ビットレート集合  $\mathfrak{B}$  は 250 から 1500kbps の 7 種類のビットレートを持ち、 $l = 2$  秒ごとのセグメントに分割されている。制御パラメータは  $T_r = 3$  秒,  $T = 10$  秒,  $\alpha = 2$  を用いた。サーバとクライアントは有線ネットワークで接続し、3G ネットワークで実測した TCP スループットの時系列データに従って、映像配信の TCP スループットを制限できるようになっている。評価結果を表 1 に示す。予測を利用した場合、映像視聴の再生停止が発生していない。平均映像ビットレートが 48.3kbps 低下しているが、TCP スループット予測により数十 kbps の低下の代わりに再生停止の回避が可能である。

#### 4.1.2 ビデオペーシング

インターネット映像配信サービスでは、ダウンロードされた映像データが視聴されずに破棄されることが多い。これは、利用されないデータがダウンロードされることは、通信キャリアに

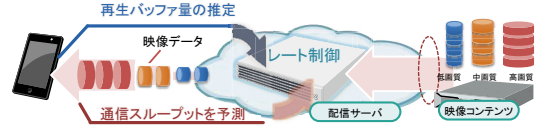


図 4 適応映像配信システム

表 1 適応映像配信評価結果

|            | 予測なし方式     | 予測方式       |
|------------|------------|------------|
| 平均映像ビットレート | 559.2 kbps | 510.9 kbps |
| 総再生停止時間    | 59.0 秒     | 0.0 秒      |

にとって通信資源の無駄になり、大きな課題になっている。

このため、映像を配信する際に送信レートを制御することでクライアントに未再生データがバッファリングしすぎないようにするビデオペーシング方式が注目されている [18]。一方、バッファリングしすぎないように制御するため、再生バッファ量が少なくなると、通信スループットが低下した場合に映像データの転送が間に合わずに再生が停止してしまう問題がある。

ここでは、通信スループット予測に基づき、映像配信の送信レートを制御することで、トラフィック削減と再生停止回避を両立するビデオペーシング方式を紹介する [19]。

ビデオペーシングのアーキテクチャを図 5 に示す。映像配信サーバにビデオペーシング機能を配備することで、映像配信サーバから送信される無駄な映像配信トラフィックを制限することができる。また、通信キャリアのネットワーク内に映像配信を中継するプロキシサーバを配置し、プロキシサーバで映像ファイルをキャッシュしながら、クライアントへの転送レートを制御することで、通信キャリア自身がモバイル網やアクセス網のトラフィックを制限することができる。

ここで紹介するビデオペーシング方式は、複数のセグメントに分割された映像データに対して配信 (on) と休止 (off) を繰り返す on-off 制御で実現する。配信区間は通信スループットに従って配信し、クライアントの再生バッファ量を目標値に近づけるように休止区間長を制御することで、送信レートを制御する。未来の通信スループットが  $x^-(t)$  に従って推移するような事象を想定した場合でも、再生バッファ量が所定値を下回らないように適応的に制御する。このように、TCP スループットが悪化する事象を考慮してペーシングすることで、トラフィック削減と再生停止回避を両立する。

シミュレーションにより、本ビデオペーシング方式の評価を実施した。映像配信サーバから本方式、従来方式を利用して、ネットワークエミュレータ経由で映像を配信し、再生バッファ量と再生バッファの枯渇による再生停止回数を計測する。ネットワークエミュレータでは 3G ネットワークで実測した通信スループットデータを利用して帯域を制限する。利用した通信スループットデータは比較的安定して高スループットが得られた 12 パターンの通信スループット (安定環境) と、通信スループット変動が大きかった 18 パターンの通信スループット (不安定環境) である。

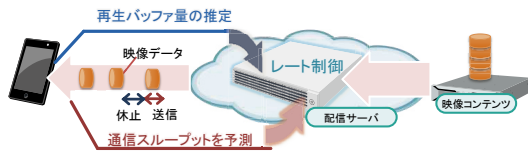


図5 ビデオペーシングシステム

表2 ビデオペーシング実験における平均再生バッファ量 (単位: 秒)

|       | 提案手法 Zippy |       |
|-------|------------|-------|
| 安定環境  | 17.88      | 31.15 |
| 不安定環境 | 25.82      | 25.88 |

表3 不安定環境における平均再生停止回数 (単位: 回)

|        | 提案手法 Zippy |      |
|--------|------------|------|
| 再生停止回数 | 1.61       | 1.78 |

比較するビデオペーシング方式は、文献[20]で紹介されている Zippy Pacing 法を用いた。Zippy Pacing 法はセグメント送信ごとに配信と配信休止を繰り返すことで、ビデオペーシングを実現する。

配信実験の結果を表2と表3に示す。表2を見ると、提案手法は従来手法に比べ、安定環境において平均再生バッファ量を40%低減できていることがわかる。ビデオペーシングによる端末の再生バッファに蓄積される量が少なくなり、視聴の取りやめによるデータの破棄が少なくなることがわかる。これに対し、不安定環境では平均再生バッファ量の低減効果はほとんどない。一方、表3は不安定環境での再生バッファ枯渇による平均再生停止回数である。提案手法は従来手法に比べ、再生停止の回数を少なくできている。なお、安定環境では提案手法、従来手法とも再生停止は発生しなかった。

以上のように、紹介したビデオペーシング手法は、通信スループットが十分得られる環境では、再生バッファ量を減らすことで、無駄に破棄されるデータ量を少なくする。一方、通信スループットが十分に得られない環境では、再生バッファ量を減らさないように制御することで、再生停止を防止することができる。

## 5. まとめ

本稿では、ベストエフォートネットワークにおけるクラウドサービスの利用品質向上を目的として、ネットワークの現在の状態をいち早く推定するもしくは未来の状態を予測するアプリケーションレベルのネットワーク状態推定・予測技術を紹介した。また、ネットワーク状態推定・予測を利用したアプリケーション例として、ネットワークモニタリング、映像配信サービスを紹介した。

## 文 献

- [1] S. Tasaka and Y. Ishibashi, "Mutually compensatory property of multimedia QoS," Communications, 2002. ICC 2002. IEEE International Conference on, vol.2, pp.1105–1111vol.2, 2002.
- [2] M. Jain and C. Dovrolis, "End-to-end available bandwidth:

measurement methodology, dynamics, and relation with tcp throughput," Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, pp.295–308, SIGCOMM '02, ACM, New York, NY, USA, 2002.

- [3] V.J. Ribeiro, R.H. Riedi, R.G. Baraniuk, J. Navratil, and L. Cottrell, "pathChirp: Efficient available bandwidth estimation for network paths," In Passive and Active Measurement Workshop, 2003.
- [4] T. Oshiba and K. Nakajima, "Quick end-to-end available bandwidth estimation for QoS of real-time multimedia communication," Computers and Communications (ISCC), 2010 IEEE Symposium on, pp.162–167, 2010.
- [5] T. Oshiba and K. Nakajima, "Quick and simultaneous estimation of available bandwidth and effective udp throughput for real-time communication," Computers and Communications (ISCC), 2011 IEEE Symposium on, pp.1123–1130, 2011.
- [6] 大芝崇, 中島一彰, "リアルタイムコミュニケーションの品質を確保するための短時間可用帯域推定方式," 情報処理学会論文誌, vol.53, no.2, pp.698–711, feb 2012.
- [7] 吉田裕志, 里田浩三, "単位根検定を用いた TCP スループットの定常性解析," 信学技報, 第 112 巻, pp.39–44, IN2012-56, 2012.
- [8] 吉田裕志, 里田浩三, "アプリケーションレベルでの TCP スループットの定常性解析と予測モデル構築," 信学技報, 第 112 巻, pp.39–44, IN2012-128, 2012.
- [9] A. Finamore, M. Mellia, M.M. Munafò, R. Torres, and S.G. Rao, "Youtube everywhere: impact of device and infrastructure synergies on user experience," Proceedings of the 2011 ACM SIGCOMM conference, pp.345–360, IMC '11, ACM, New York, NY, USA, 2011.
- [10] Apple Inc., "HTTP live streaming overview," 2011.
- [11] A. Zambelli, "IIS smooth streaming technical overview," March 2009.
- [12] Adobe Systems Inc., "HTTP dynamic streaming on the adobe flash platform," 2010.
- [13] T. Stockhammer, "Dynamic adaptive streaming over HTTP -: Standards and design principles," Proceedings of the second annual ACM conference on Multimedia systems, pp.133–143, MMSys '11, February 2011.
- [14] S. Akhshabi, A.C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," Proceedings of the second annual ACM conference on Multimedia systems, pp.157–168, MMSys '11, ACM, New York, NY, USA, 2011.
- [15] 吉田裕志, 里田浩三, 野垣智士, "TCP スループットの確率的拡散予測に基づく映像配信制御," インターネットコンファレンス 2011 論文集, pp.57–66, 2011.
- [16] 吉田裕志, 里田浩三, "TCP スループット予測技術の映像ストリーミング制御への応用とその評価," 信学技報, 第 112 巻, pp.281–286, IN2012-201, 2013.
- [17] 吉田裕志, 里田浩三, "映像パケットの流体モデルに基づく映像ストリーミング制御," 信学技報, 第 108 巻, pp.411–416, IN2008-201, 3 月 2009.
- [18] K.J. Ma, R. Bartoš, S. Bhatia, and N. Raj, "Mobile video delivery with HTTP," Communications Magazine, IEEE, vol.49, no.4, pp.166–175, April 2011.
- [19] K. Satoda, H. Yoshida, H. Ito, and K. Ozawa, "Adaptive video pacing method based on the prediction of stochastic TCP throughput," Global Communications Conference (GLOBECOM), 2012 IEEE, pp.1944–1950, 2012.
- [20] K.J. Ma, R. Bartoš, and S. Bhatia, "Scalability of HTTP pacing with intelligent bursting," Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on, pp.798–801, June 2009.