



## ユーザに優しいインターネット 酒井・山岡 研究室～集積システム専攻



山岡 克式 准教授

酒井 善則 教授

インターネットが発達、普及するにつれて、送受信されたり蓄積されたりするデータの量は急激に増加している。しかし、インターネットの回線は世界中に張り巡らされているため、機器の改良などによる通信性能の向上が困難になっている。

酒井・山岡研究室では、通信環境を向上させてこのような状況に対処するために、効率の良い情報処理の論理や仕組みを考案している。主にネットワークの制御技術と、マルチメディアの処理方法を総合的に研究しているが、本稿ではその研究の一部について紹介する。



### ネットワークシステムとその活用

自分の生涯を写真とともに残して紹介するブログや、世界中から投稿されたあらゆる動画を保存し公開するサービスなどがインターネット上には存在している。これらの利用者は増え続け、今では世界中のサーバに保存されているデータの総量が非常に多くなっている。

データ量の増加だけでなく、使用されている回線の速度も以前と比べて格段に上がっている。以前は東工大でも、インターネットといえばほとんどがメールの送受信であったため、回線速度が大学全体で 64 Kbps でも困ることはなかった。ところが現在では、インターネットで扱われるファイルの形式も多様化し、動画像などデータ量が非常に多いファイルの送受信も頻繁に行われるようになったことから、10 Gbps まで上げられている。

回線に対する需要が増えていることから、将来供給の不足が起こりうる。そこで、通信機器を全て高速な光回路素子や光回線によるものにしてしまおうという話もある。しかし、それには膨大な時間と費用を必要とするため実現は困難である。さらに既存のソフトウェアが、改良したハードウェアと互換性を持たず受け入れられない場合も

ある。そこで酒井・山岡研究室では、ハードウェアではなく、通信方法及びマルチメディア制御などのシステムやアルゴリズムを改良することで、効率のよい通信方式の実現を目指している。

酒井・山岡研究室の扱う研究内容は幅広い。教授が学生に研究テーマを提案するだけではなく、学生が自分でやりたいテーマを考案し、独自に研究する場合もある。

例えば、データの通信単位であるパケットを、送信元で可能な限り集めひとまとめにした後インターネットへ送信する、パケットマージと呼ばれる手法の研究を研究室では行っている。データがインターネット上を移動する際には、複数の送信元と受信先が同時に一本の回線を利用できる必要がある。全てのコンピュータ同士の間に一本ずつ回線を用意することは、非現実的かつ無駄が多いからだ。そのため、インターネットでは送信したいデータを細かく分割し、それぞれに送り先の情報を付けた「パケット」を用いるパケット通信を行う。このパケット通信によって、同じ一本の回線を通していても各パケットの行き先が区別できるため、同時に複数の通信が可能になる。

このパケットをどこに送るのか決定してその通りに送信する装置をルータと言い、コンピュータやサーバなどと接続されている。パケットがルータを通過する際、ルータはパケットの送り先を一つ一つ確認することになる。従来では回線の余裕の有無にかかわらず、細かいパケットを大量に処理していた。しかしデータ量が多い場合、確認すべきパケットの数も多くなり、通信に時間がかかる。一方パケットマージを用いると、ルータを通過する一つ一つのパケットのデータ量は多くなるが、送り先の情報を確認する回数は少なくなり通信時間を短縮できる。さらに、この手法を実際のインターネットに取り入れる際も通信のシステムを変えるだけで済み、回線を取替えたり増やしたりする必要はない。

他にも、ウェブコンテンツ推薦システムの研究が行われている。ウェブのユーザは、自分が公開

している画像や音楽などのコンテンツに、タグと呼ばれる簡単な言葉をつけてその内容を区別している。コンピュータがコンテンツを検索する際、従来はタグの比較のみで検索を行っていた。しかしこれでは、似たような内容のコンテンツであってもタグの言葉が異なると見つけれず、ユーザは欲しい情報が得られなくなる可能性がある。研究室では、タグだけではなくコンテンツの内容そのものがどれだけ似ているのかということも比較して、目的のコンテンツを探していく方法を考え出した。この研究では効率のよいコンテンツの探し方を提案している。この推薦システムによって、ユーザが求めているコンテンツをコンピュータがより速く推薦できるかもしれない。

酒井・山岡研究室では、他にも多様な研究が行われているが、本稿では画像の圧縮、ファイル配送システム、パケット交換制御について紹介する。



## 逆 Colorization による画像の圧縮

近年、携帯電話を含めたデジタルカメラの画素数は向上し、撮影した写真をメールに添付して送信したり、ブログで公開したりするケースが増えている。インターネット上では写真などの画像が縦横無尽に流れているが、一度に送信できるデータ量には限界がある。インターネット上で大量に画像を扱うには、画像データを圧縮し、回線に負担をかけないようにしなければならない。

画像データは、明るさを表す輝度情報と、明るさを無視した色情報に大きく分けられる。人間は輝度に敏感であり、色情報はあまり差異を感じないため、色情報についてはデータを多少減らしても構わない。この理由により、JPEG など現在幅広く利用されている画像の圧縮方式では、色情報の一部を省略することによってデータを圧縮している。酒井先生をはじめとした研究チームは、次に述べる Colorization を用いて色情報の圧縮を行う手法を考えた。

輝度情報だけで表現されている画像を、グレースケール画像と呼ぶ。このグレースケール画像に色情報を与える手法として、Colorization というものがある。写真など通常の画像は、輝度が近い場所では色も近くなることが多い。この仮定を前提として、輝度情報、ユーザが与えた色情報(色

の付いた線)により、コンピュータで色を広げる手法が Colorization である(図1)。

色情報を表す線分の色と位置が予め適切に指定されている場合は、Colorization によって元の画像とほぼ同じものを復元できる。酒井研究室では、Colorization を復元ではなく圧縮に応用しようと考え、新しい手法を確立しようとしている。その手法が「逆 Colorization」である。

以前に他の研究者が Colorization に似た方式で画像を圧縮しようと考えていたが、その研究では線ではなく大量の点を用いて圧縮しようとしていたため、あまり効率よくデータ量を減らすことができなかった。点より広い部分をカバーできる線

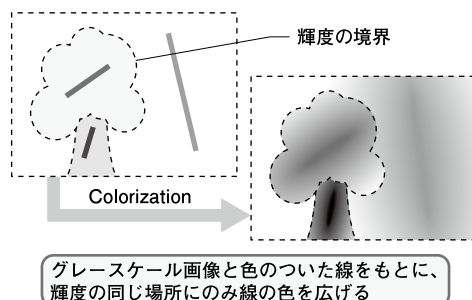


図1 Colorization による色の広がり方

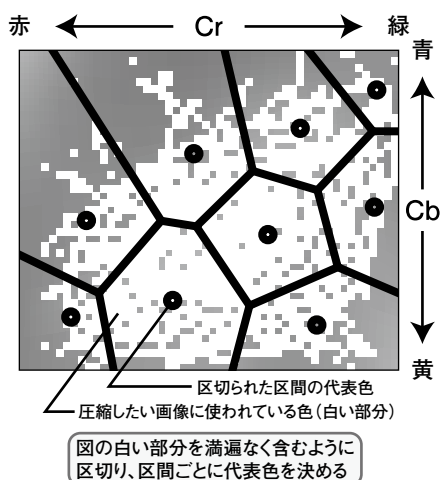


図2 色差空間

を用いることが Colorization の特徴である。この特徴を利用して、酒井先生の研究チームでは圧縮率を上げることを考えた。

さらにデータ量を減らすために、研究チームでは「代表色」という概念を導入した。代表色とは、それらを混ぜ合わせることで、画像内の全ての色を不足なく表現できるように選ばれた色である。代表色の数を少なくすると、画質はやや下がるがデータ量は少なくなる。この代表色のみを用いて線を引いていく。

代表的な色を表す成分として、赤、緑、青の光の強さを表す RGB がある。一方、逆 Colorization では RGB の光の三原色の軸を変換した YCbCr という別の色成分を用いて代表色を決定する。ここで Y は輝度、Cb 成分は青色に近いが黄色に近いを表す度合を示し、Cr 成分は赤色に近いが緑色に近いを表す度合を示す。Y 成分はグレースケールを表すため、代表色として使う色は、Cb、Cr 成分からなる輝度が均一な色に限る。

最初に、圧縮したい元の画像に使われている色を抽出する。次に、図2の通り使われている色が含まれるよう、使う代表色の数だけ色差空間を細かい区間に分割する。色差空間とは、Cb と Cr の各成分を、それぞれ軸とした平面のことである。色差空間の Cb 座標と Cr 座標をそれぞれ指定することにより、色情報で使用する任意の色を選択できる。

元の画像に使われている色が表現できるように分割した各区間から代表色を選ぶ。塗るべき範囲

を大きくカバーするように、Colorization で用いる色の線分を代表色だけで引く。ある一つの代表色で線を引いている間、他の代表色は全て塗られているものとする。単に一本引いただけでは、コンピュータが色を広げた時に元の画像との誤差が大きいため、さらに線分を引いて修正する。元の画像との誤差がユーザの決めた許容範囲以内になったら、次の代表色に移る。あとはこの操作を、全ての代表色に対して繰り返す。代表色にない色は、線分の位置に応じて異なる代表色を重ね合わせることによって作り出す。

逆 Colorization によって圧縮された画像を、データの相違を計算して圧縮前の画像と比較する手法で評価すると、その差は大きくなってしまいます。線分という限られた情報量を用いて改めて色を付け直したため、デジタルな観点で比較すると元の画像の色情報がほとんど残っていないのだ。しかし写真のような画像を評価するのは最終的には人間であり、人間にとってきれいに見えればそれで十分と考えるなら、全く問題はない。

そこで、一度逆 Colorization で圧縮し、再び復元した写真を複数の人に見てもらい、その品質を人の目で5段階評価する「主観評価実験」を行ってみた。すると、JPEG で圧縮した画像より少ないデータ量でも高い評価を得ることができた。この結果は画像によって差があるものの、JPEG に劣ることはなく、データ量を更に少なくした画像では、評価の差がより顕著になっている(図3)。

元々 JPEG は、データを画像に復元する計算量を減らすために開発された方式である。JPEG が

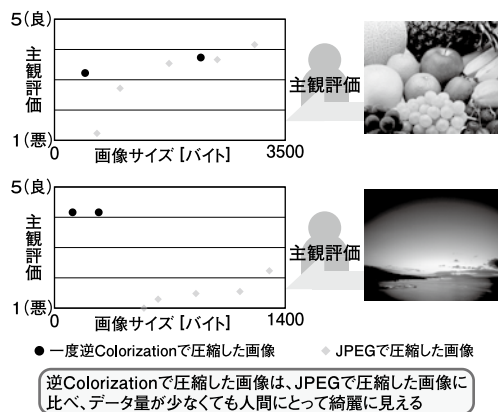


図3 主観評価実験の結果

考え出された当時は、コンピュータの処理能力がまだ低かったため、少ない計算量で復元を行えることを目的としたのである。Colorization に関しても、復元では色の線を輝度に応じて広げるだけなので、少ない計算量で行うことができる。

ところが、JPEG による画像の圧縮がほぼ一瞬で終わるのに対し、逆 Colorization で圧縮する場合は、 $400 \times 400$  ピクセル (画素) の画像だと数秒かかってしまう。その理由として、線を引くたびに元の画像との差が最小になるよう計算しなければならないことがある。差を最小にするためには、Colorization によって画像が正確に復元されるようにしなければならない。輝度が似ている場所では色も似ているという仮定の下に Colorization は行われているため、一つのピクセルに対して、他のピクセルの輝度と色情報を考慮しながら重み付

けをしていくことになる。ピクセルの比較の計算は、行列を用いた連立方程式によって行う。先で述べた  $400 \times 400$  ピクセルの画像の場合、計 16 万ピクセル一つ一つの色の重み付けを他のピクセルそれぞれについて計算しなければならないため、約 16 万  $\times$  16 万の正方行列を係数にもつ連立方程式を解くことになるのだ。

従来、画像データを圧縮する側は、主に設備の整った放送局などの企業や団体であり、復元する側は、携帯電話や小型のデジタルカメラなど計算量が限られた機器を持つ、家庭や個人であった。しかし最近では、個人でも画像を作成、送信する機会が多くなっている。そのため、大掛かりな機器を使うことなく短時間で圧縮を可能にすることが、この逆 Colorization における一番の課題となっている。



## ファイル配送システム

ダウンロードしたファイルの複製物を、一時的に複数のコンピュータに保存するウェブプロキシ、パソコン同士で直接データを共有する P2P など、現在では多様な方式のファイル配送システムが利用されている。この配送システムは、複製物を回線で接続された隣のコンピュータにも置く「キャッシュ機能」により、サーバやルータ等の機器にかかる負荷を分散、削減している。それでも一つのファイルにリクエストが集中することがあり、求めるファイルもユーザによって異なるためにファイルを受け取るまでに時間がかかることがある。ここではファイルをリクエストするユー

ザの「満足度」をいかに向上させられるかということに主眼を置き、理想的なファイル配送システムを目指す学生の研究を紹介する。

ここで言う「満足度」とは、ユーザ全体の平均のサービス時間がどれだけ短いかということであらわす尺度である。サービス時間とは、ユーザが欲しいファイルのリクエストを送信してから配送が完了するまでの時間のことで、ファイルのデータ量や回線の状況などによって異なる。

回線を共有している個々の人が別々のファイルのリクエストを送信している場合、適切な配送順序は、個々のファイルのデータ量に依存する送信所要時間や、リクエスト数によって決まる。例えば図 4 のようなモデルでは、4 人のユーザが送信元のコンピュータにファイルをリクエストする。

ファイルのデータ量が全て同じで、リクエスト数が異なる場合、リクエスト数が多いファイルを先に送った方が良い。この場合、人気の低い、つまりリクエスト数の少ないファイルを受け取る人のサービス時間は長くなってしまいが、そのような人は少数である。一方、人気の高いファイルを受け取る大多数の人のサービス時間は短くなる。すると、結果としてユーザ全体の平均サービス時間の短縮、すなわち満足度が向上する。

リクエスト数が同じ場合、データ量の少ないも

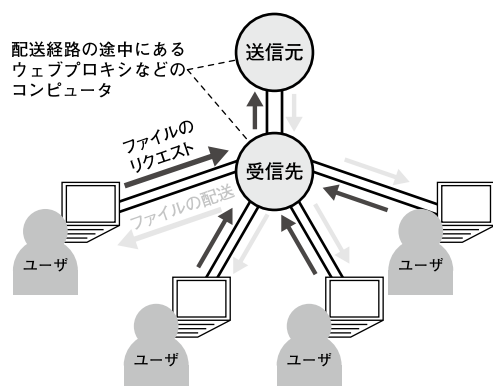


図 4 ファイルの配送モデル



のから送信することで、満足度が上がる。データ量の少ないファイルの送信は早く終わるため、次に送るファイルの待ち時間が短くなり、結果として合計のサービス時間が短縮されるからである。

実際には、送信するそれぞれのファイルのデータ量もリクエスト数も異なり、これから新たに追加されるリクエストも予測できない。それではどのような優先順位を付けて送れば良いのか。配送しようとしているファイルをそのまま先に送った方が良いのか、あるいは他のデータの優先順位を高くし、配送しようとしているファイルの前に割り込ませて配送させた方が良いのか。



## TACCS によるパケット交換の制御技術

データを細かく分割し、送り先の情報を付けてインターネットへ送るパケット通信では、性質上一連の通信が完了して初めて意味のあるデータとなる。パケット通信の流れを便宜上「フロー」と表現することがある(図5)。簡単なテキストファイルの送受信一件におけるパケットの流れも、音声や動画などをダウンロードしながら再生するストリーミングの動画一本分のパケットの流れも、一つのフローとして扱う。

ルータにはパケットの送信先を確認したり、実際に送信したりする配送処理の能力に限度があるが、処理されるデータの種類は動画、テキストなど様々である。種類が異なると扱うパケットの数も異なることから、ルータでパケットが混雑するタイミングが掴みづらい。もし回線が混雑してルータの処理能力の限度を超えた場合、処理しきれなかった分のパケットはそのルータで全て破棄される。この現象を「パケットロス」という。パケットロスが発生すると、少しでもパケットを失ったフローは全て途切れてしまうことになる。現在一般的に使用されているルータは、限度を越えた分のパケットを、所属するフローに依らず無差別に破棄してしまう。そのため、一度のパケットロスで途切れるフローが多くなり、必要以上に破棄される情報が多くなる。文字数の少ないテキストなど、小さなファイルの送受信ならば再送も簡単にでき、受信側も特に困ることはない。一方ストリーミングでは、途中でパケットロスにより動画のパケットが一部失われてしまうと、その失

その答えは、「(リクエストの数)÷(残り配送時間)」を「優先度」と見て、その優先度の高い順に送る方法である。簡単な結果だが、これが最適解であることは、数学的帰納法を用いて証明できる。実験でも、追加されるリクエストがいつ来るか事前に分かっている場合と、かなり近い結果が出た。

この方法が最適解となるのは、全ての回線やコンピュータが等しい処理能力を持つ場合に限る。実際のインターネットは研究室で使用したネットワークよりも複雑で規模が大きいため、実用化に向けて改良が続けられている。

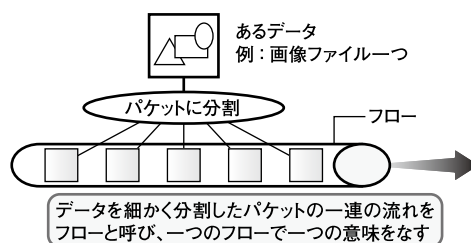


図5 パケットとフロー

われた分のパケットを再送することになる。しかし再送が遅れてしまうと、到着が間に合わなかったパケットに相当する画像の一部が抜けたまま送信先で再生されてしまうため、サービスの質が低下することになる。ストリーミングのようなリアルタイムの通信を行う場合は、従来のように多数のフローが無差別に途切れてしまうと多くのユーザーが困る。これを防ぐために、送信途中でパケットロスの被害を受けないようある程度のルータの処理能力を確保させ、フローが最後まで途切れないように送信しなければならない。そこで山岡研究室で考案された手法が、TACCSである。

今までは、ルータの処理能力から、そのルータが受信、配送しているデータの単位時間の量を差し引いて、回線に追加して入れられるフローのデータ量を計算していた。しかし差し引いているのは、あくまで配送しているデータ量の単位時間当たりの「平均」であり、実際には使用量が状況に応じて、短時間で大きく上下する。そのため計算上送信できるはずのデータ量のフローを入れた場合でもパケットロスが起こり得る。また、新し

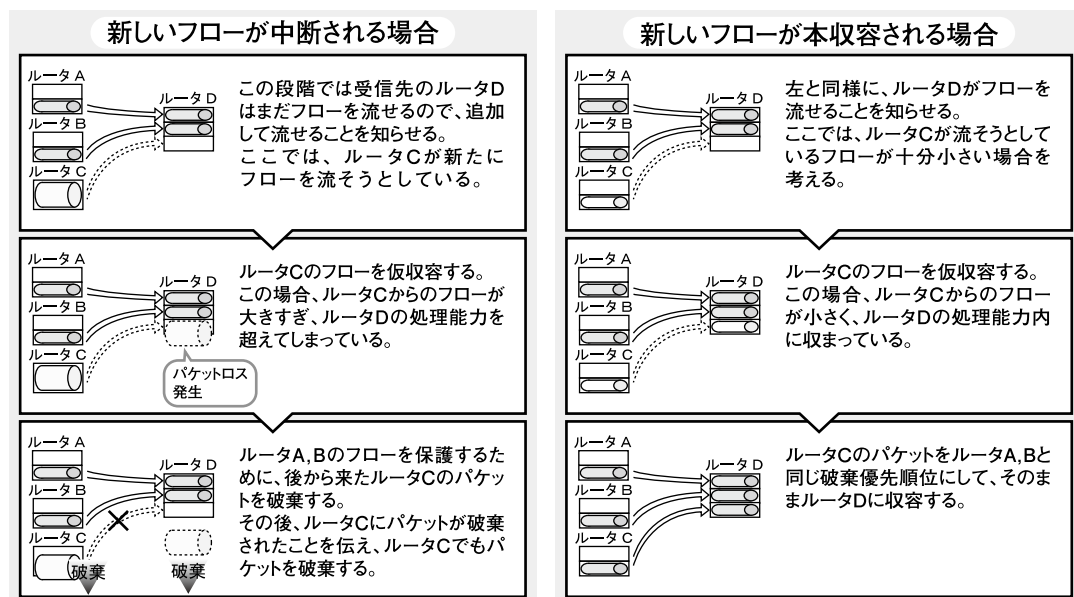


図6 TACCSの手順

いフローを入れられたとしても、そのフローを入れた時点から、回線に追加して入れられるデータ量を計算し直すのはルータにとって負担となる。TACCSはフローを単位としてパケット通信を制御することで、ルータに上記のようなデータ量の計算をさせることなく、パケットロスの影響を最低限にすることを目的としている。以下では、TACCSの具体的な手順を紹介する(図6)。

まず、送信先のルータDの処理能力に少しでも余裕がある場合、送信元のルータCに対して、ルータDに追加してフローを流せることを通知する。ルータCは、流そうとしているフロー内のパケットに破棄される優先順位を設定し、そのフローはルータDに一時的に収容される。これを仮収容という。新しいフロー内のパケットに設定された優先順位は、すでにルータDに収容されているルータA、B由来のパケットに設定された順位よりも高い。一方ルータA、Bのフローは既に収容されているため、A、B由来のパケットが破棄される優先順位は同じである。フローを仮収容すると、フロー内のパケットはルータDへ送られる。

新しく到着したフローが仮収容されてパケットが送られた結果、ルータDが処理能力を超えそうになったことを自ら検出したとする(図6左)。その場合、ルータDは仮収容したフローに属するパケットを送信せずに破棄する。その後ルータC

にパケットが破棄されたことを通知し、ルータCでもそのフローのパケットを破棄することで、フローが中断される。仮収容されてもルータDの処理能力に余裕があり、パケットロスが起きる可能性がなければ、ルータDは仮収容したパケットの優先順位を先に収容したパケットと等しくして、そのまま本収容する(図6右)。

このTACCSを用いると、パケットロスが発生しても破棄されるものは仮収容したフローに属するパケットのみであり、すでに本収容された分は保護される。仮収容したフローがパケットロスを起こして中断されたとしても、本収容した他のフローが巻き添えになることはない。さらに送信先のルータは、送信元のルータにフローを追加して流せるか否かを伝えればよく、処理できるデータ量まで計算する必要もない。パケットロスが発生しそうになった場合、仮収容したフロー内のパケットを全て破棄し、そのフローのみ中断させることがTACCSの一番の特徴である。

しかし現時点では、TACCSはインターネットなどへの適用が困難であることが分かっている。インターネットをはじめとした大規模なネットワークには、ルータの中でもネットワークの中心に位置し、他の多くのルータと繋がっているコアルータが多い。コアルータにはパケットの配送のみに処理能力を集中させることが望ましく、パ

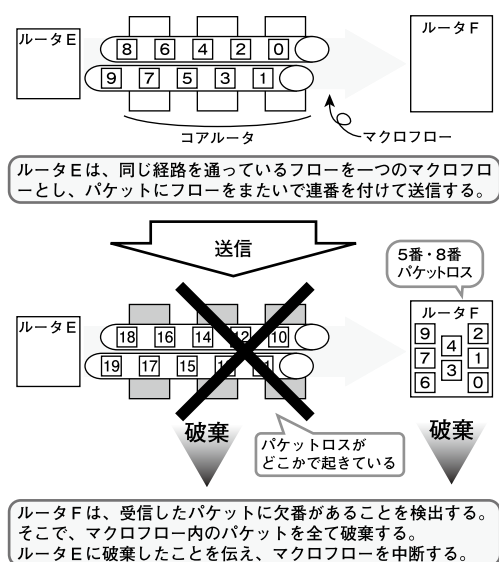


図7 edge based TACCS

ケットロスの検出をさせるほどの能力を持たせるためには、一つ一つのルータを改良して機能を向上させなくてはならない。ただ、検出作業をさせるためにコアルータを全て改良しようとすると、非常に大きなコストがかかってしまう。

TACCSの問題を解決するには、コアルータではなく、インターネットの端に位置する簡単に交換可能なルータだけに検出作業をさせればよい。その考えをTACCSに取り入れた手法がedge based TACCSである。この方法の特徴は、パケットロスが起きたルータを確認せずに、原因となるフローを制御することである。

まず、同じ経路を通る複数のフローをまとめて一つの「マクロフロー」として考える(図7)。送信元のルータEは、個々のフローをまたいで、マクロフローのパケットに0、1、2、……と番号を付けていく。収容したマクロフロー内のパケットを送信した結果、送信先のルータFで受信したパケットに欠番が見つかったとする。これはこのマクロフローが通った経路の中のどこかのコアルータで、パケットロスが起きたことを示す。この場合、ルータFはTACCSの場合と同じよう

にそのマクロフロー内のパケット全てをルータFで破棄し、ルータEに通知する。

この一連の作業によって、パケットロスが起きている経路を検出したことになる。そして、その経路内のどこかにある、パケットロスを起こしたコアルータの通信量をフローの中止により緩和させることが出来る。もちろん、コアルータの機能には何も変更を加える必要がない。

この方式では、途中のコアルータで起きたパケットロスそのものは制御できず、パケットが送信先のルータに到着するまでパケットロスが起きたかどうかは分からない。そのため、複数のマクロフローが通過しているコアルータでパケットロスが発生した時、破棄されたパケットを含む一部のマクロフローが中断されたとしても、別のマクロフローでは属するパケットが全て通過し、中断されない恐れがある。これではパケットロスを全て検出できず、コアルータの負荷を完全に軽減できない。山岡研究室では、この「棄却ミス」の与える影響についても計算及び実験を行った。

結果、棄却ミスの与える影響は、実用には問題ない程度であることが分かった。というのも、棄却ミスが起きるのは、中心のルータでパケットがあまり破棄されない時、言い換えるとそのルータを通過するパケットが少ない時である。このような場合は、そもそもパケットロスがほとんど起こらないため、棄却ミスがあっても回線が急に混雑することはなく実用上は問題がない。逆にパケットロスが起ころうなほどルータが混雑している場合は、途中のコアルータで多くのパケットが破棄される。すると、送信先のルータではパケットロスを多く検出しフローを適切に中断させることができるため、棄却ミスが起これにくい。そのことが計算結果と実験結果から示されたのである。edge based TACCSは、元のTACCSよりも実用性を持った方式であると言える。

このように酒井・山岡研究室では、情報通信の仕組みや手順を改良することで、実社会でも利用可能なネットワーク機能の向上を目指している。

通信についての知識をほとんど持っていなかった私にとって、今回の取材は通信システムへの関心を大いに抱くものとなりました。

最後になりましたが、お忙しい中重度重なる取材に快く応じて下さった酒井先生、山岡先生、研究室の方々に心より感謝いたします。(金刺 宏樹)