大規模トラフィックの解析技術

濱口　佳孝　　伊加田　恵志
中村　信之

近年ネットワークに流れる通信トラフィック量は増大し続けている。今後も音声や動画のストリーミングデータが伸び、さらにセンサ情報等の新たな通信やサービスの出現に伴いこの傾向は続くと思われる。現在でもネットワークの管理のためにこれらのトラフィックの観測・解析が行われているが、その処理対象は年々増加しているといえる。

特に国内でのサービスが始まったNGNでは、各ネットワーク事業者が機器の動作状況を直接管理できる個々のネットワーク内の品質保証のみではなく、ネットワークを跨いだ通信での品質保証も求められるため、通信トラフィックの監視が必要となる。たとえば**図1**に示したように、複数のネットワークを跨り端末間でRTP（Real-time Transport Protocol）による通信が行われる場合、①通信の品質情報を端末がRTCP（RTP Control Protocol）を用いて互いに送信するので、②網間でそのRTCPの情報を収集し、③ネットワーク品質を解析することなどが必要と考えられる。このように、各ネットワーク事業者にとってネットワークを跨る通信が集中する箇所、すなわち網間に置かれるセッションボーダーコントローラー（SBC）等でそこを流れる通信トラフィックを観測・解析することで、ネットワーク品質を監視し、品質劣化の要因発生部分の切り分けを行うことが重要になる。このような解析の実現のためには、事業者網など大規模なネットワーク同士の網間での膨大な通信トラフィックという、大規模な時系列データを効率よく解析する手法が必要となる。

本稿ではこのような大規模な時系列データの解析手法として、大量の通信の中から障害に関係があるものを選び出すために用いられる頻出値抽出手法と、普段のトラフィックの状況からの予測と現実の測定値との差から異常を検出するために用いる学習手法についての我々の取り組みを述べる。

時系列データの解析の従来研究

本節では、時系列データの解析についての従来研究を紹介する。データ群の傾向の分析や学習･予測を行う手法としてデータマイニングという分野が良く知られている。しかしデータマイニングの手法では計算にデータを複数回使う必要があるものや、データ数が増えると計算コストが急激に増大するものが多い。このような手法を通信トラフィックのような時系列データに対して適用すると、次のような問題が発生する。まず、データマイニング処理

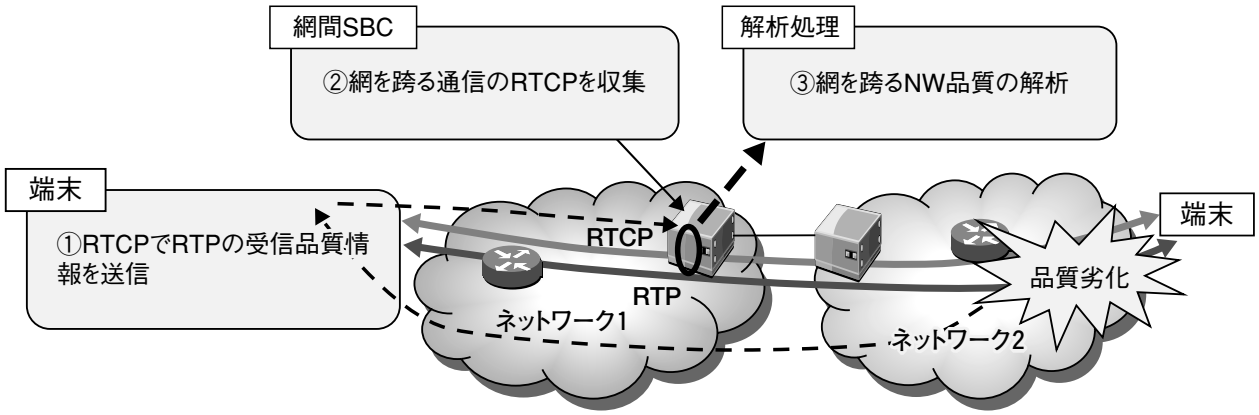

図1　網間でのトラフィック観測による、ネットワーク品質推定の概念

を行う間はデータをメモリ上に保存しておく必要があり、大きなメモリを必要とすると共に、次から次へと来るデータの管理が複雑になる。さらに、到着するデータ量に対して処理時間がスケーラブルでないために、データの流量が増加すると、次のデータの処理が間に合わなくなる場合がある。

これらの問題を解決するために、近似等によりある程度の計算誤差を許容することでデータマイニングの処理の工夫を行う、ストリームマイニングと呼ばれる分野の研究が近年行われるようになってきている。この手法では、データは1度処理した後は使わないようにすることでメモリ量を減らし、計算量もデータ量に対して比例する程度とすることを実現している。

本稿で紹介する頻出値抽出については、Manku[1]らがLossy Counting アルゴリズムを提案している。このアルゴリズムは誤差 ε を許容し、頻出値抽出にその誤差未満の影響しか与えないデータの出現数を記憶しないことで、データの種類が多くなっても ε により決まる少ないメモリ量での処理を実現できる。このように少ないメモリに時系列データの解析に必要な情報のみを収めたものをスケッチと呼ぶ。たとえばパケットロスが頻出する送信側と受信側のIPアドレスの組を抽出しようとした場合、その組み合せの数は膨大になり、ぞれぞれのIPアドレスの組についてのパケットロスの出現数を保持するためには大きな記憶容量や複雑な管理が必要になる。しかしこのアルゴリズムを用いれば、頻出する一部のIPアドレスの組についてのみ出現数と誤差の情報をスケッチとして記憶するだけで所望のIPアドレスの組が抽出できることになる。

ネットワーク監視への適用における課題

一方、従来のストリームマイニングの研究はデータマイニング手法の延長にあり、多くの場合、与えられたデータは一定の特徴を持つことを前提としてその特徴を抽出することを目的としている。すなわち、新たなデータが来るごとに解析結果を逐次的に更新し続け、データの特性が変化するポイントをリアルタイムで検知するようなことには向かない。そのため、そのままではネットワークの監視には使い難い。

たとえば先に紹介したLossy Counting アルゴリズムで統計的に意味のある一定期間Nの間に頻出する値を抽出し、それより短いサイクルでその抽出値を更新したいとする。そうすると**図2**に示したように複数の測定区間が並列することになり、その並列した測定区間分の処理量と記憶容量が必要となる。つまり、頻出値のような統計量が変化する所をリアルタイムに捉えることでネットワークの監視をしようとすると、処理コストが大きくなり、ストリームマイニングのメリットを受けることが難しくなる。

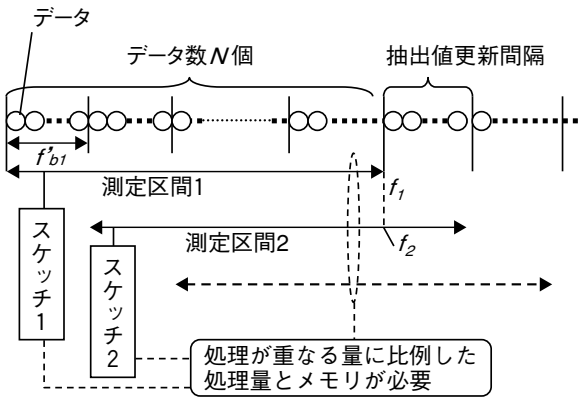

図2　Lossy Counting アルゴリズムによる逐次処理

頻出値抽出手法の拡張

一定期間中のIPアドレスごとの品質劣化の回数を数え、品質劣化が頻出するIPアドレスを大規模な通信トラフィックから抽出するために、我々は、前述のLossy Counting アルゴリズムをネットワーク監視に用いる上での課題を解決した拡張手法を開発した[2]。本節では、その手法の概要を述べる。

基本的なアイデアは、**図2**の測定区間1の処理が終わった時点のスケッチ1から、その時点での測定区間2のスケッチ2を近似的に算出することにある。これが実現すれば、測定区間2の次の区間についてもスケッチ2から算出することで並列して処理を行う必要がなくなり、測定区間より短いサイクルでの頻出値抽出結果の更新が可能となる。Lossy Counting アルゴリズムのスケッチには、ある程度以上の頻度で出現するデータそれぞれについて、その出現頻度 f と、頻度の誤差に相当する Δ が保持されている。測定区間1が終了した時点での測定区間2の頻度 f_2 は、測定定区間1での頻度 f_1 と、測定区間1の開始位置から測定区間2の開始位置までの出現頻度 f'_{b1} を用いて、$f_2=f_1-f'_{b1}$ とすればよい。ここで問題は、f'_{b1} の情報はスケッチにはなく、また、もし各区間においてこれを記憶しておくと結局メモリを大きく消費してしまうことにある。

一方、ストリームマイニングは誤差を許容することで処理量やメモリの削減を目指すものであり、Lossy Counting アルゴリズムも ε の誤差を許容している。すなわち、f_1 や f_2 は正確である必要は無く、値が誤差 ε（測

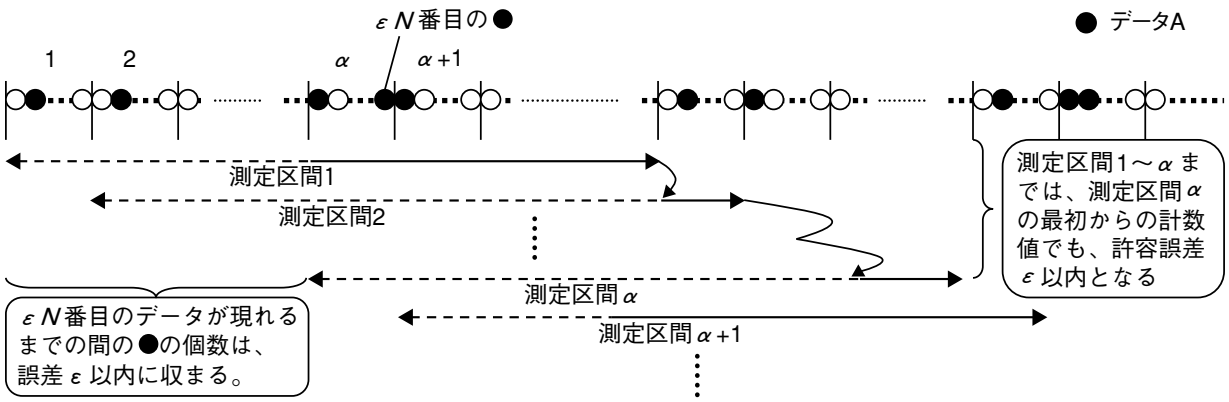

図3　頻出値抽出の改良手法の概念

定区間のデータ数がNなので、頻度ではεN個）以内に収まれば良いことになる。例として**図3**に示したように、あるデータAについて測定区間1の最初からεN個目の出現位置が、測定区間のずらし幅でα番目の位置の場合を考える。この場合、測定区間1の開始位置から、測定区間αの開始位置の間にあるデータAの個数はεN個より小さいため、この出現数を無視しても許容誤差εの範囲に収まる。すなわち測定区間1～αについては、測定区間αの最初の位置からの出現数で近似できるため、その値をそのまま使う。測定区間α＋1以後についてはα＋1を新たな測定区間1として同じ処理を繰り返せば良い。

まだ測定区間が若干重複するように見えるが、このアイデアの処理は全データに対して行う必要はなく、元のアルゴリズムでスケッチに記憶されたデータに対してのみ処理すればよいため影響は小さい。これについて従来手法では**図2**のように測定区間が50区間重複するような処理のシミュレーション実験を行った結果、従来手法では処理の重複数に比例して単一の処理の約50倍の処理時間がかかるのに比べて、本手法では約2%の処理時間増加にすぎないことを確認した。このアイデアで処理を組み替えることで、品質劣化が多発するIPアドレスの組を継続的に観測した通信トラフィックから抽出し続けることが可能になる。

時系列データの学習・予測手法

本節では、時系列データの学習・予測を行い、予測結果と実際の観測データとの乖離から異常の検知を行うことを目指した技術について述べる。

動画配信等でのトラフィック予測については、Liang[3]により、多重解像度解析の結果をニューラルネットワークで学習することがVBR（Variable Bit Rate）での通信に効果があることが示されている。これを実際のトラフィック監視に適用するためには多重解像度解析の処理量がデータ量に対して比例程度に収まり、逐次処理が可能でなくてはならない。これについては、乱数を用いた情報の圧縮を利用して、多重解像度解析としてのウェーブレット解析をストリームマイニング化する手法がGilbert[4]らにより報告されている。

我々は、これらの手法を組み合せ、RTPの流量と、それと相関が高いSIP（Session Initiation Protocol）のセッション数を用いた学習によりRTPの流量を予測させる実験と評価を行った[5]。ただしGilbertらの手法では、ウェーブレット係数の近似精度があるデータ長までしか保証できない。トラフィック監視に用いるためには継続的にウェーブレット係数の算出を続ける必要があるため、今回の実験では近似が保証できるデータ長に達するよりも学習処理に必要なデータ長分だけ前から、並列して改めて乱数による情報の圧縮を開始するようにしている。

実験に使用した通信トラフィックデータは、7つの網を接続した状態を想定したネットワークをシミュレータ上に構築し、OKI内で実際にIP電話等の通信で観測したデータを元に作成した通信シナリオで通信トラフィックを発生させたものである。この通信トラフィックから5秒ごとにSIPのセッション数とRTPの流量を統計値として取り出し、本手法を用いて学習を行い、その学習結果を用いてRTPの流量予測を行った。その予測値と実測値を**図4**に示す。

ニューラルネットワークの入力としてウェーブレット解析の結果を使わなかった場合、予測値と実測値に大きな乖離が出る部分があった。しかし、ウェーブレット解析を組み合わせた場合は、それが乱数による情報圧縮を

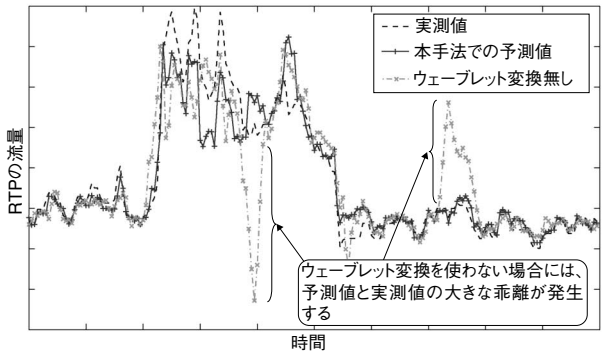

図4　RTP流量の予測値と実測値

行った近似値であっても、その乖離は改善されていた。この手法により、予め時間をかけて学習した結果を用いて予測をするのではなく、ストリームマイニング化されたウェーブレット解析との組み合せによりにより常時リアルタイムに学習結果を更新しながら予測を行うことができるようになる。

ま と め

本稿では、時系列データの解析に用いられるストリームマイニングに対して、時系列データの特徴が変化する場合に対応するために、測定区間をずらしながら解析結果を出力するにあたり、処理に必要なCPUやメモリを大幅に削減する手法を述べた。また、学習・予測手法にストリームマイニングを組み合わせることでスケーラブルに学習・予測を行う手法について、RTP流量の学習と予測により本手法の効果の検証を行った。今後、このようにストリームマイニング技術を具体的に通信トラフィックのリアルタイム解析によるネットワーク監視に適用する研究を進め、通信トラフィック流量が増加しても、商品として実用的なハードウェアと処理時間での処理を実現する、通信トラフィック解析技術を開発していく。

TIPS【基本用語解説】

RTP：Real-time Transport Protocol
音声、映像等のリアルタイム通信に用いられるプロトコル。

RTCP：RTP Control Protocol
RTPの通信の調整のために、受信端末から送信元へ受信品質情報等の送信を行うプロトコル。

SIP：Session Initiation Protocol
IP電話等で、相手の呼び出しや通信の開始・終了等の制御を行うプロトコル。

VBR：Variable Bit Rate
通信の状況に合わせてビットレートを変化させる方式。そのため、セッション数とトラフィック量が比例しない。

謝　辞

本研究は、情報通信研究機構（NICT）の委託研究「次世代ネットワーク（NGN）基盤技術の研究開発」の一環として実施したものである。 ◆◆

■参考文献

1）G.S.Manku and R.Motwani：“Approximate frequency counts over data streams”, Proc. VLDB'02, pp.346–357, 2002
2）伊加田，濱口：効率的な頻出データ計数アルゴリズム Lossy Counting の拡張，電子情報通信学会信学技報，Vol.107 No.524，pp.43–47，2008年
3）Y.Liang：“Real-Time VBR Video Traffic Prediction for Dynamic Bandwidth Allocation”, IEEE Trans. Systems, Man, and Cybernetics Part C, Applications and Reviews, Vol.34 No.1, 2004
4）A.C.Gilbert, Y.Kotidis, S.Muthukrishnan and M.Strauss: “Surfing Wavelets on Streams: One-Pass Summaries for Approximate Aggregate Quaries”, Proc. of CLDB, pp.79–88, 2001
5）伊加田，中村，濱口：NGNにおけるネットワーク異常検出のためのRTPトラフィック予測手法，電子情報通信学会信学技報，Vol.109 No.79，pp.67–72，2009年

●筆者紹介

濱口佳孝：Yoshitaka Hamaguchi. 研究開発センタ ユビキタスシステムラボラトリ
伊加田恵志：Satoshi Ikada. 研究開発センタ ユビキタスシステムラボラトリ
中村信之：Nobuyuki Nakamura. 研究開発センタ ユビキタスシステムラボラトリ