

圧縮センシングを用いた時空間階層化トラヒック観測手法

辻 喜宏[†] 大下 裕一[†] 村田 正幸[†] 山崎 晃嗣^{††} 宮崎 昭彦^{††}

[†] 大阪大学 大学院情報科学研究科 〒565-0871 大阪府吹田市山田丘 1-5

^{††} 日本電信電話株式会社 NTT 先端集積デバイス研究所 〒243-0198 神奈川県厚木市森の里若宮 3-1

E-mail: [†]{y-tsuji,y-ohsita,murata}@ist.osaka-u.ac.jp, ^{††}{yamazaki.k,miyazaki.akhiko}@lab.ntt.co.jp

あらまし 時間変動の大きなトラヒックを収容するには、ネットワーク中に流れるトラヒック量をリアルタイムに把握することが必要となる。そのためには、短い時間間隔で多量のトラヒック情報をネットワークを介して収集することが必要で、特に、観測器から転送されたトラヒック情報が集中するネットワーク管理サーバに近いリンクには大きな負荷をかけてしまう。本稿では、圧縮センシングの技術を応用した、リアルタイムなトラヒック情報を把握することができる手法を提案する。提案手法では、まず、観測器-管理サーバ間で事前に生成し共有したトラヒックモデルからの差分データのみを圧縮し送信することを考える。圧縮データを収集する上では、事前に従った経路に従って圧縮データを一つのパケットに集約して送信していくことで、管理サーバ付近の帯域を圧迫する問題を解決する。本稿では、数量的なシミュレーション評価によって、提案手法の有効性の評価を行う。

キーワード トラヒック観測、トラヒック収集、圧縮センシング、トラヒックモデル

Hierarchically Structured Spatio-Temporal Traffic Measurement utilizing Compressive Sensing

Yoshihiro TSUJI[†], Yuichi OHSITA[†], Masayuki MURATA[†],

Koji YAMAZAKI^{††}, and Akihiko MIYAZAKI^{††}

[†] Graduate School of Information Science and Technology, Osaka University

1-5 Yamdaoka, Suita-shi, Osaka, 565-0871 Japan

^{††} NTT Device Technology Laboratories, NTT Corporation

3-1 Morinosato-wakamiya, Atsugi-shi, Kanagawa, 243-0198 Japan

E-mail: [†]{y-tsuji,y-ohsita,murata}@ist.osaka-u.ac.jp, ^{††}{yamazaki.k,miyazaki.akhiko}@lab.ntt.co.jp

Abstract In the traffic engineering, we need a real-time traffic information. The real-time traffic information is obtained by periodically collecting the traffic information from traffic monitors. However, collecting a large amount of traffic information takes a large overhead to the network, especially at the links near the control server. In this paper, we propose a method that can collect the real-time traffic information without causing a large overhead to the networks. In this method, we first create the traffic prediction model and share it between the traffic monitor and the control server. Then, the difference from the model is collected. When collecting the differential data, the data is compressed based on the compressive sensing. In this paper, we evaluate our method by simulation and demonstrate that our method can collect traffic information without causing a large overhead.

Key words Traffic Measurement, Traffic Data Collection, Compressive Sensing, Traffic Modeling

1. はじめに

時間変動の大きなトラヒックを収容するには、各時刻のトラヒックに合わせて経路を変更するトラヒックエンジニアリングと呼ばれる手法が有効である。従来、トラヒックエンジニアリ

ングによる経路制御は1日周期の時間変動に合わせて1時間から数時間間隔での検討が進められてきた[1],[2]。しかしながら、ネットワークを流れるトラヒックの時間変動が大きくなると、より短い周期での制御を行うことが必要となる。文献[3]では、データセンター内において、サーバ間に十分な通信帯域を確保

するためには、秒周期で経路制御を行うことが必要とされることが明らかにされている。

トラヒックエンジニアリングにおいて、各時刻における適切な経路を計算するためには、その時刻のネットワーク内の全地点間を流れるトラヒック情報を把握することが必要となる。トラヒック観測を行う方法として、従来から Netflow [4] や sFlow [5] といった技術が広く用いられている。NetFlow では、観測器においてフローごとにカウンタを用意し各フローの統計情報をカウントし、管理サーバが定期的にフロー統計情報をネットワーク内の全観測器から収集することにより、トラヒック情報の把握を行う。それに対して、sFlow では、サンプリングされたパケットの情報を集約し、直接管理サーバに送出する。そして、管理サーバではサンプリングされたパケットの情報をもとに、トラヒック情報の把握を行う。しかしながら、制御周期が短くなると、観測データの収集も短い周期で行うことが必要となる。その結果、多量のトラヒック情報をネットワークを介して収集することが必要となり、特に、観測器から転送されたトラヒック情報が集中するネットワーク管理サーバに近いリンクには大きな負荷をかけてしまう。トラヒック観測に関して、さまざまな研究が行われている [6]~[8] もの、それらは、観測器において多量のトラヒックをフロー単位に分類して観測を行うことを目的としており、観測器から管理サーバへのトラヒック情報転送にかかる負荷については、十分に議論されていない。

そこで、本稿では、この問題に対して、ネットワークに負荷をかけず、リアルタイムなトラヒック情報を把握する手法を提案する。提案手法では、まず、ネットワーク観測器とネットワーク管理サーバで、トラヒック予測モデルを共有する。各タイムスロットにおいて、各観測器は、予測モデルからの誤差が定められた値以上となるトラヒックエントリの情報についてのみ、量子化した後にネットワーク管理サーバに送ることになる(時間方向の圧縮)。このとき、予測モデルからの差分データを送信する際には、圧縮センシングの考え方を応用し、情報を収集する経路にそって、圧縮しながら送ることにより、管理サーバ付近のネットワークの帯域を圧迫することを防ぐ(空間方向の圧縮)。また、提案手法では、観測情報を収集する上で、量子化、圧縮センシング(圧縮・復元)によって生じる誤差を、トラヒックエンジニアリングに影響ない範囲に抑えながらも、観測情報収集にかかるオーバーヘッドを削減する方法を紹介する。

本稿の構成は次の通りである。まず、2. 章では圧縮センシングのトラヒック観測への応用方法(収集情報の生成方法・収集情報からの復元方法)を説明する。次に、3. 章では観測情報の誤差と収集のオーバーヘッドを考慮した収集制御方法を説明する。4. 章ではモデルデータを用いた評価によって、提案手法の有効性を確認する。最後に、5. 章では結論と今後の課題についてまとめる。

2. 圧縮センシングを利用したトラヒック観測

2.1 圧縮センシング

本稿では、圧縮センシングをトラヒック観測に応用すること

を考える。圧縮センシングは信号・画像処理の分野で注目を浴びている技術で、データに含まれる冗長性を利用したサンプリングを行うことによってデータサイズの削減を図るものである。圧縮センシングでは、原信号 $x \in \mathbb{R}^N$ と観測信号 $y \in \mathbb{R}^M$ の間に $y = Ax$ の関係式が成り立つ場合に、観測信号 y から原信号 x を推定する問題を取り扱う。この問題は、劣決定系の(解よりも式の方が少ない)連立方程式を解く必要がある不良設定問題であり、通常解が定まらない。しかし、観測行列 A のコヒーレンシーが小さく、原信号 x がスパースである(=非ゼロ要素の数が少ない)場合に、わずかな誤差で復元できることが知られている [9]。圧縮センシングにおいて、圧縮率と復元率との間にはトレードオフの関係が存在するが、それらの特性は、観測行列の設定方法や各復元アルゴリズムに対して明らかにされている [9]~[12]。本稿では、圧縮センシングにおける原信号を各観測器で枚タイムスロット毎に発生するトラヒック観測情報とし、観測信号を圧縮した収集対象情報を観測信号として管理サーバまで収集するものとする。トラヒックエンジニアリングで必要となる観測情報には誤差として一定の値を許容できることを利用し、その範囲以内に観測情報に含まれる誤差を抑えつつも収集におけるオーバーヘッドを減らすことを考える。

一般的に、圧縮センシングにおける圧縮・復元の性能はデータのスパース性に依存すると言われており、スパース性が高い信号(あるいは、基底変換を行うことによって得られるスパースな信号)適用することが考えられる。そのため、本稿では、圧縮センシングをトラヒック観測へ応用するために、各観測器で圧縮対象となる原信号のスパース性を確保する必要がある。本稿では、図 1 に表されるような、トラヒックデータの符号化・復号を考えることによって、トラヒック観測への圧縮センシング適用を可能とする。まず、トラヒックデータのモデル化を行う。トラヒックのモデル化が正確に行えれば、各時刻に観測されたトラヒックデータとトラヒックモデルの差をとることにより得られるトラヒック差分データは高いスパース性を示すことが期待できる。事前にそれらのモデルデータを送信側・受信側で共有しておけば、収集が必要な観測情報はトラヒック差分データであり、これらのデータに圧縮センシングを適用することによって収集のオーバーヘッドを大幅に削減できると期待できる。

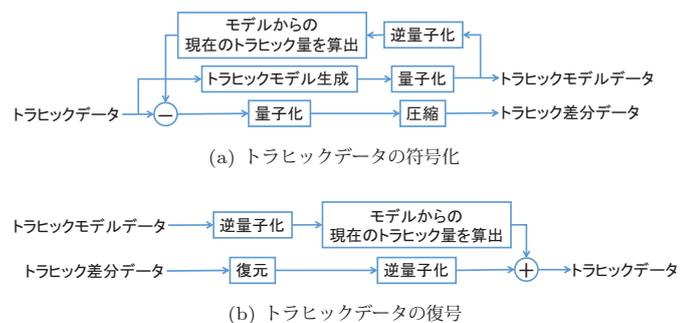


図 1 トラヒックデータの符号化・復号

2.2 トラヒックデータの圧縮と復元

各観測対象のフローに対して、トラヒック予測モデルを構築

し、観測情報送信側と受信側で共有をする。そして、観測情報送信側では、観測したトラフィック情報が予測モデルとの誤差が閾値を超えた場合に、モデルとの差分情報を受信側に伝える。この差分情報を用いることにより、受信側では正しいトラフィック情報を把握することができる。また、予測が正確に行える場合、各時刻において、誤差が閾値を超えるフローエントリ数は限られると考えられる。そのため、差分情報は、圧縮センシングの考え方を用い、圧縮して送信することができる。以降、本節では提案手法におけるトラフィック情報の生成・圧縮・復元方法について述べる。

2.2.1 量子化

観測情報送信側では、まず、圧縮センシング適用対象となるデータのスパース性を確保するため、予測値からの差分データは事前に定められた量子サイズによって量子化する。フロー f に該当する差分データ x_f は以下のように表される。ただし、 T_f はフロー f の実際に観測された値、 \hat{T}_f は予測モデルから得られる予測値、 ϵ_f はそれらの差分値、 ρ は量子化サイズであり、 $\text{trunc}(x)$ は 0 からの距離 $|\text{trunc}(x)|$ が $|x|$ を超えない最大の整数のうち x と同じ符号を持つ整数を表す (例: $\text{trunc}(0.4) = 0, \text{trunc}(-1.2) = -1$)。

$$x_f = \text{trunc}\left(\frac{\epsilon_f}{\rho}\right) = \text{trunc}\left(\frac{T_f - \hat{T}_f}{\rho}\right) \quad (1)$$

2.2.2 圧縮

圧縮差分データ y は各フローエントリ f_i の量子化後差分データ x_{f_i} を用いて表される $x = (x_{f_1}, \dots, x_{f_n})^t$ と観測行列 A を用いて、次のように求められる。このときの原信号 x のサイズ n に対する観測信号 y のサイズ m の縮小比 n/m が圧縮率に相当する。

$$y = Ax \quad (2)$$

本稿では、文献 [12] で述べられている手法を用い、コヒーレンシーが小さくなるよう、以下の値を最小化することによって観測行列 A を定める。コヒーレンシーを小さい観測行列 A を設定することによって、より広範囲のスパース性な信号に性能が良い圧縮センシングが適用可能になることが知られている。

$$A = \arg \min_A \|A^t A - I\|_F^2 \quad (3)$$

2.2.3 復元

観測情報受信側では、圧縮された差分データ y から、各フローの差分データ \hat{x} を復元する。本稿では、最も一般的な再構成アルゴリズムである、OMP (Orthogonal Matching Pursuit) アルゴリズムを用いて差分データの復元を行う。 \hat{x} は以下の式により得ることができる。

$$\hat{x} = \arg \min_x \|y - Ax\|_2^2 \quad \text{s.t.} \quad \|x\|_0 \leq \epsilon \quad (4)$$

2.2.4 逆量子化

復元された差分データ \hat{x} は量子化サイズ ρ を用いて逆量子化される。その後、得られた各フローの差分データとフローの予

測値を用い、各フローの現在のトラフィックデータの推定値 \hat{T}_f を次のように取得することができる。

$$\hat{T}_f = \rho \hat{x}_f + \hat{T}_f \quad (5)$$

3. 時空間階層化トラフィック観測

3.1 概要

本稿では、観測システムは図 2 に表されるような動作を行い、トラフィック観測情報がネットワーク帯域を圧迫する問題を解決する。本システムにおいては、1) 予測と連携して予測誤差が閾値以下のエントリについては、管理サーバに何も情報を生成せず、2) 予測誤差が閾値を超えたエントリについてのみ、予測値からの差分データを情報収集の経路にそって 1 パケットに圧縮して送る。提案手法では、各観測器において、過去に観測されたトラフィック情報をもとに、学習対象の各フローエントリについて、トラフィック予測モデルを生成し、生成された予測モデルを管理サーバに送出する。以後、各観測器においては、各フローについて、生成された予測モデルを用い、予測との誤差を計算する。そして、誤差があらかじめ定められた許容値を超える場合は、予測値との差分データを生成する。各観測地点で生成された差分データは、管理サーバを根とする木にそって送出される。その際に、各観測器では、自身の子となる観測器から転送されてきた差分情報を含むパケットと自身の差分情報を合わせて 1 パケットとなるように圧縮した上で転送を行う。この方法により、ネットワーク内のいずれの地点においても、各観測周期ごとに 1 パケットしか流れず、ネットワークの帯域を圧迫することを防ぐことができる。管理サーバでは、送られてきた差分情報パケットに対して、復元処理を行い、各フローエントリの差分情報を得る。その後、各フローの予測モデルと当該差分情報を足し合わせることで、当該時刻のトラフィック情報を得る。特に、観測情報の効率良い収集を実現する上での、観測器間の連携方法を次に詳しく説明する。

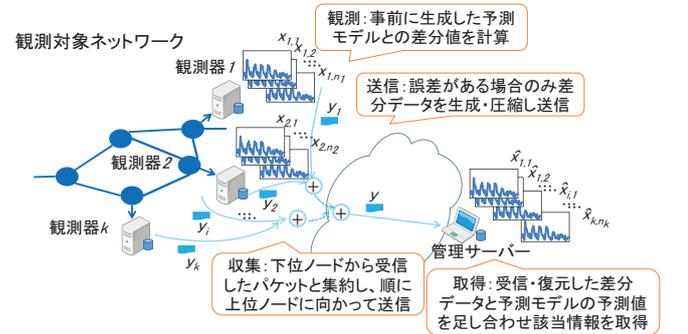


図 2 観測システムの動作概要

3.2 階層的なトラフィックデータの収集

本稿では、各観測地点で発生した、スパースである差分データを 1 パケットに集約して送る方法として、CDG (Compressive Data Gathering, [13]) を検討している。CDG は圧縮センシングされたセンサ情報の低帯域な収集方法であり、文献 [13] では、

大規模センサネットワークにおけるセンサ情報収集に応用されており、各観測地点で生成された観測信号を、収集経路上で圧縮・集約していくことによって、1パケットによる観測情報の収集が可能となる。本来は、異なる観測地点で別空間の観測信号が生成されるところを、原信号を観測地点を超えて同空間に定義することによって、観測システム全体で一つの観測信号の定義が可能となる。観測器 i において生成される圧縮データ y_i は、木構造で形成される差分データ収集経路において自身の子となる観測器から送られた圧縮データと、自身のフローの観測結果 x_i をもとに以下のように行われる。

$$y_i = Ax_i + \sum_{j \in C_i} y_j \quad (6)$$

ただし、 x_i は i で観測されないフローについては 0 としたベクトルであり、 C_i は i の子ノードの集合である。このよう階層的に形成された収集経路上の各観測器で圧縮・集約を行うことによって、管理サーバ付近のリンクの帯域使用率を削減することができると思われる。

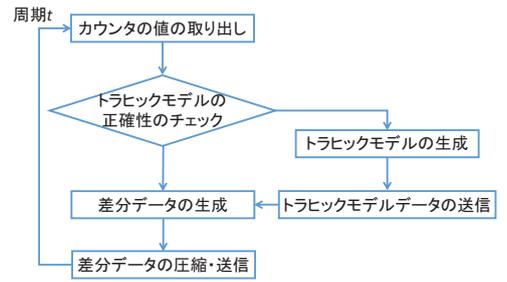
3.3 トラヒックモデルの更新

観測器と管理サーバ側で、同じモデルを共有することが出来れば、収集に必要な観測情報は差分データのみであり、量子化によって得られるそれらのデータはスパースであることが期待できる。しかし、遠い将来の予測が難しいように、一般的に遠い将来であればあるほど、トラヒックのモデルによって取得できる予測値と観測されるトラヒックデータとの差分データは大きくなり、量子化によって得られる差分データのスパース性は低下することが考えられる。データのスパース性低下によって、圧縮センシングの性能が悪化し、観測誤差を許容範囲以内に抑えるために必要な観測信号のサイズ拡大を引き起こしてしまう。この事態を防ぐために、観測器と管理サーバの動作としてモデルの正確性のチェック、更新の処理が必要で、それらの処理を含めた観測システムの各構成要素の動作は図3のように表される。各フローエントリの真の値は、観測器しか知り得ないので、トラヒックモデルの正確性のチェックは観測器が行う。トラヒックモデルの更新の必要性の判定は、様々な判定ルールが考えられる。たとえば、そのフローエントリが連続 λ のタイムスロットで送信必要な差分データが存在する場合に、トラヒックモデルの更新を行うといった判別手法が考えられる。

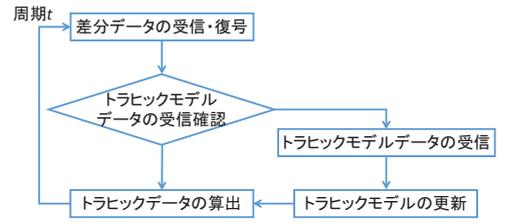
4. 評価

4.1 評価環境

予測誤差モデルを用いてモデル化以降の一連の手順に関して、観測情報に含まれる誤差(量子化、圧縮センシング)を許容範囲する範囲に抑えつつ、観測情報収集における計算量をどれだけ削減可能であるかを調査する。予測誤差を変化させてそれぞれの観測環境において、提案手法による削減可能帯域を評価する。本評価において、予測誤差には次の二種類の誤差が存在するものとし、それらの特徴を持つフローの割合を変えることによって突発的な変動が頻繁に含まれる/含まれない観測環境(予測モデルによる予測値が正確な場合/正確でない場合)を実



(a) 観測器の動作



(b) 管理サーバの動作

図3 観測システムの各構成要素の動作

現するものとする。1) 値が小さすぎて、予測モデルでは予測しきれない僅かな誤差(例: ホワイトノイズ)と、2) 突発的な変動により、予測モデルでは予測しきれない大きな誤差(例: スパイク)。

観測ネットワークの評価環境は以下の通りである。各観測地点 i で観測される回線は 1Gbps の使用率を記録し、100(= n_i) 個観測対象フローが存在する(平均フロー量 $\mu = 10$ [Mbps]) のものとする。また、各タイムスロットに発生する予測モデルからの誤差 ϵ はガウス分布 $N(0, \sigma^2)$ に従うと仮定し、予測モデルでは予測しきれない、小さな誤差と大きな誤差はそれぞれ標準分散として $\sigma_1 = 100$ [Kbps], $\sigma_2 = 3$ [Mbps] を持つものとする。各フローエントリ f の予測誤差 ϵ_f は $N(0, \sigma_1^2)$, $N(0, \sigma_2^2)$ のいずれかに従うが、観測器 i で観測される大きい予測誤差を持つ分布に従うフローの数(= n_spikes_i) を $[0, 50]$ の範囲で変化させることによって様々な観測環境を実現する。量子化サイズ ρ を $[0.01, 2.50]$ [Mbps]、圧縮サイズ $|y|$ を 50, 25(それぞれ圧縮率 2 倍、4 倍) に設定した場合、提案手法による削減可能計算量を調査する。

また、提案手法の一連の動作をシミュレーション評価する上で、本稿では、文献 [12] で示される観測行列の最適化アルゴリズムを用いて生成された観測行列 A を、原信号の復元アルゴリズムとして OMP アルゴリズムを採用する。観測行列 A を生成する上で、引数として、 $step_size = 0.02$, $Iter = 100$ を与える(各引数の説明は文献 [12] を参照)。復元アルゴリズムは、Python の機械学習ライブラリ scikit-learn [14] で提供される `sklearn.linear_model.OrthogonalMatchingPursuit` 関数を用いて実装を行う。

4.2 評価結果

予測誤差モデルに含まれるそれぞれのスパイク数に対して、観測誤差、すなわち、量子化・圧縮センシングによって生じる誤差を評価した結果を図4、5に示す。図4、5はそれぞれ圧縮率 2

倍、4倍の場合の評価結果を表す。本評価では、平均誤差・最悪誤差は、式(5)で得られる各フローエントリ f の推定値 \hat{T}_f を用いて、それぞれ、 $\frac{1}{n} \sum_{i=1}^n (|\hat{T}_{f_i} - T_{f_i}|)$, $\max_{i=1, \dots, n} (|\hat{T}_{f_i} - T_{f_i}|)$ と定義される。いずれの圧縮率でも、観測誤差の値のグラフは v の字の形を描くことが確認できる。誤差 $|\hat{T}_{f_i} - T_{f_i}|$ は各フローエントリ要素で発生した量子化による誤差が圧縮センシングによって拡大された値となる。量子化サイズが小さい場合は、差分データの非ゼロの要素は多いために、圧縮センシングの処理で生じる誤差が大きく占める。量子化サイズが大きい場合は、差分データの非ゼロの要素数は少ないために、圧縮センシングの処理で生じる誤差は少なくなるものの、量子化により発生する誤差が大きくなる。この量子化による誤差は量子化サイズが要素の最大値(平均値)を超えると、それ以上は発生し得ない。観測誤差としての最小値は、圧縮センシングによる誤差と量子化による誤差の影響の均衡が取れた場合であり、量子化サイズが 0.25 [Mbps] の値付近のときに確認できる。

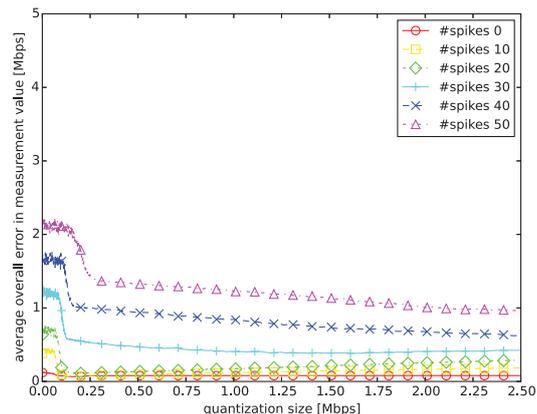
最悪誤差・平均誤差を 1 [Mbps] 以内に抑えるに圧縮率が2倍の図4の場合には正確なトラヒックモデルが得られていないフローの数は5割程度、圧縮率が4倍の図5の場合には3割程度まで許容可能であり、圧縮率が高くなると、許容誤差の範囲以内に収めるために、予測モデルから外れたフローの数は小さく抑える必要がある。図4の場合、スパイク数が圧縮サイズの5割(全フロー数での比率では2割5分)に収まる観測環境においては、量子化サイズを0.25 [Mbps] に設定することによって、各差分データを送る場合に比べて50%の観測情報量で1 [Mbps] の誤差以内でトラヒック把握が可能である。

4.3 モデル更新に関するパラメータ

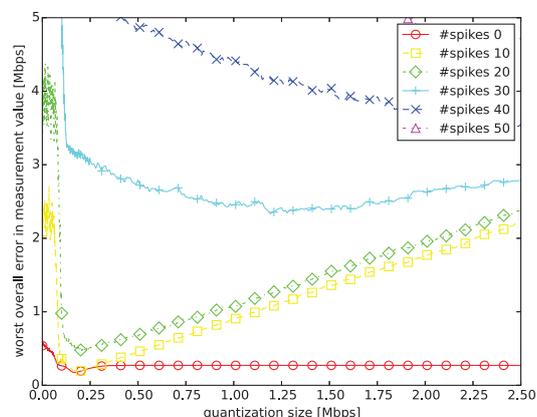
4.2章では、モデルからの差分データを圧縮することで観測が必要となるデータ量の削減量を評価した。しかし、実際は遠い将来を予測することが難しいように、時間方向でのモデルの性能悪化を考慮した制御を行う必要がある。圧縮率50%、観測情報の許容誤差を1 [Mbps] に抑えたい場合には、量子化サイズを0.25 [Mbps] に設定すればよい。このとき、予測誤差が大きなフローの割合を全フローの2割5分程度に抑える必要がある。よって、予測モデル作成直後の最高性能時に各時刻において1割のフローに大きな誤差が含まれるとし、1トラヒックモデルモデルが時々刻々悪化し大きな誤差が含まれるフロー数がタイムスロットが進むにつれて前のタイムスロットの1.1倍になるとすると、 $1.1^\lambda < 0.25/0.1$ を満たす最大整数値 $\lambda = 9$ タイムスロット毎にモデルの更新をしてやればよい。この制御のためには、全体のトラヒックデータに含まれる誤差の大きなフロー数を各観測器が把握する必要があるが、各観測地点で自身が観測するフロー数に対してあらかじめモデル性能が悪化し大きな誤差が生じるフロー数に対して閾値を決めておけば十分であると考えられる。

5. まとめ

本稿では、トラヒックデータを収集する上で、時間モデルからの差分データを圧縮センシングを用いて圧縮し、観測器間で



(a) 平均誤差



(b) 最悪誤差

図4 観測情報に含まれる誤差 ($|y| = 50$ 、圧縮率2倍)

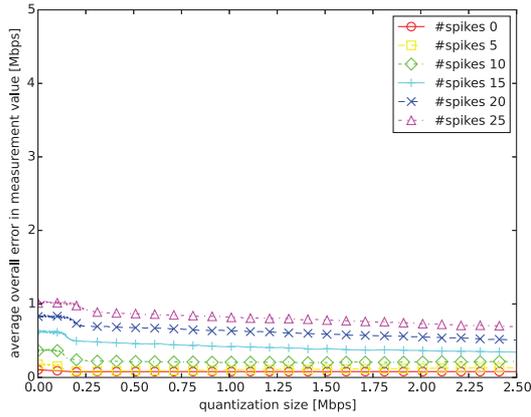
集約していくことで、観測情報収集のオーバーヘッドを削減できる手法を提案した。

評価の結果、誤差が大きなフロー数を2割5分に抑えることができれば、観測情報の誤差を1 [Mbps] 以内で収集のオーバーヘッドを50%削減可能なことを示すことができた。

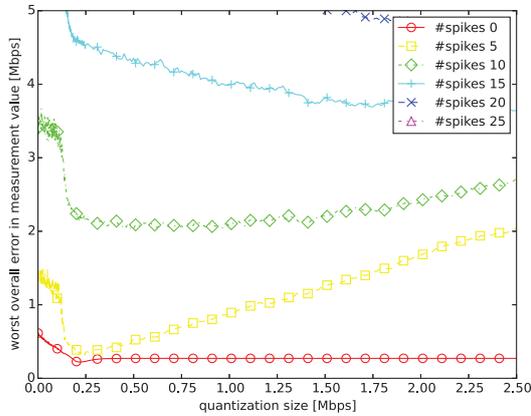
今後の課題として、提案する観測手法実現のために、トラヒックモデル生成手法と連携させた評価が考えられる。また、フローの空間相関性を利用し、さらなる観測負荷の削減を行う手法も今後検討を行う予定である

文 献

- [1] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg, "COPE: Traffic Engineering in Dynamic Networks," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4, pp. 99–110, Aug. 2006.
- [2] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering," RFC 3272, 2002.
- [3] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine Grained Traffic Engineering for Data Centers," in *Proceedings of ACM CoNEXT*, 2011, pp. 8:1–8:12.
- [4] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954, Oct. 2004.
- [5] P. Phaal, S. Panchen, and N. McKee, "InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched



(a) 平均誤差



(b) 最悪誤差

図 5 観測情報に含まれる誤差 ($|y| = 25$ 、圧縮率 4 倍)

- and Routed Networks,” RFC 3176, 2001.
- [6] G. Cormode, F. Korn, S. Muthukrishnan, and D. Srivastava, “Diamond in the Rough: Finding Hierarchical Heavy Hitters in Multi-dimensional Data,” in *Proceedings of ACM SIGMOD ICMD*, 2004, pp. 155–166.
- [7] L. Jose, M. Yu, and J. Rexford, “Online Measurement of Large Traffic Aggregates on Commodity Switches,” in *Proceedings of USENIX Hot-ICE*, 2011.
- [8] L. Yuan, C.-N. Chuah, and P. Mohapatra, “ProgME: Towards Programmable Network Measurement,” *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 97–108, Aug. 2007.
- [9] Y. C. Eldar and G. Kutyniok, *Compressed Sensing: Theory and Applications*. Cambridge Univ. Press, 2012.
- [10] J. Tropp and A. Gilbert, “Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit,” *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec 2007.
- [11] M. Elad, “Optimized Projections for Compressed Sensing,” *IEEE Transactions on Signal Processing*, vol. 55, no. 12, pp. 5695–5702, Dec 2007.
- [12] V. Abolghasemi, S. Ferdowsi, B. Makkiabadi, and S. Sanei, “On Optimization of the Measurement Matrix for Compressive Sensing,” in *Proceedings of European Signal Processing Conference*, 2010, pp. 427–431.
- [13] C. Luo, F. Wu, J. Sun, and C. W. Chen, “Compressive Data Gathering for Large-scale Wireless Sensor Networks,” in *Proceedings of ACM MobiCom*, 2009, pp. 145–156.
- [14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.