

令和 5 年 6 月 22 日現在

機関番号：12612

研究種目：基盤研究(C)（一般）

研究期間：2019～2022

課題番号：19K11938

研究課題名（和文）プロトコルの詳細な振舞いを考慮した機械学習によるTCPトラヒックの解析・制御

研究課題名（英文）Analysis and Control of TCP Traffic by Use of Machine Learning Considering Protocol Behaviors

研究代表者

加藤 聡彦（KATO, Toshihiko）

電気通信大学・その他部局等・名誉教授

研究者番号：90345421

交付決定額（研究期間全体）：（直接経費） 3,300,000円

研究成果の概要（和文）：再帰型ニューラルネットワーク（Recurrent Neural Network：RNN）に基づく識別器を用いて、ネットワークを流れるパケットを監視した通信ログから、送信側のTCP輻輳制御アルゴリズムを推定する方法を研究した。識別器に与えるデータとしては、送信側の輻輳ウィンドウの値の時間変化を推定した値を用いた。隠れ層が1層の単純なRNNと、3段の隠れ層を用いたディープラーニングによるDRNN（Deep RNN）を用いた。代表的な10の輻輳制御方式に適用し、十分な精度で推定可能であることを示した。さらに、インターネットに接続されたサーバの輻輳制御アルゴリズムの分布を推定した。

研究成果の学術的意義や社会的意義

TCPはインターネットを流れるトラヒックの主要部分を占める。その振る舞いは輻輳制御アルゴリズムにより決定づけられる。このため現在インターネット上でどのようなアルゴリズムがどの程度使用されているかを推定することは重要となる。本研究は、ネットワークを流れるトラヒックの通信ログに基づいてその送信側の輻輳制御アルゴリズムを推定する方法を提案するもので、機械学習を用いて、最新の輻輳制御アルゴリズムまでの推定を可能としたものである。ローカル環境での評価で正しい推定が可能であることを確認し、さらにインターネットに接続された実際のサーバのアルゴリズムの推定を行ったことは、十分に学術的および社会的な意義がある。

研究成果の概要（英文）： We studied a method to estimate the TCP congestion control algorithm on the sender side from a log of packets flowing over the network, using a recurrent neural network (RNN)-based classifier. As the data given to the classifier, we used a temporal change of the congestion window value on the sender side.

A simple RNN with one hidden layer and a DRNN (Deep RNN) based on deep learning with three hidden layers were used. We applied our classifier to 10 widely-used congestion control schemes and showed that it is possible to estimate with sufficient accuracy. Furthermore, we estimated the distribution of congestion control algorithms for servers connected to the Internet.

研究分野：ネットワーク工学

キーワード：TCP 輻輳制御 機械学習 ニューラルネットワーク モニタリング

1. 研究開始当初の背景

研究開始当初においても、機械学習を用いたネットワークの解析・制御については多くの研究事例が報告されていた。しかしその多くは、実測データをそのまま機械学習アルゴリズムに適用するという手法を採用していた。そこで、本研究計画では、TCP トラヒックの解析・制御を対象として、TCP プロトコルの詳細な振る舞いを考慮し、機械学習による分類・推定に適した特徴付データを抽出し、そのデータを機械学習アルゴリズムに適用するという方法を提案した。

特に、トラヒック解析の例として、通信ログから TCP の輻輳制御方式を自動的に推定する方法について研究を進めることとした。輻輳制御方式は TCP トラヒックを特徴づけるものである。このため、ネットワーク事業者にとっては、自分の運営するネットワークにおいてどのような輻輳制御方式がどの程度利用されているかを調査することは重要である。しかし輻輳制御方式は TCP のデータ送信側の内部動作として実現されているため、外部での推定は困難である。提案者は 2016 年から 2019 年の科研費で、輻輳制御アルゴリズムの推定方式を研究した。しかしこの方法は人手による推定に依存していたため、その自動化が望まれていた。そこで、機械学習を用いた自動的な輻輳制御アルゴリズムの推定方式を実現する必要があった。

2. 研究の目的

本研究は、パッシブに収集された通信ログから、ログに含まれる TCP セッションが使用している輻輳制御方式を、機械学習を用いて推定する方法を確立することを目的とする。機械学習の手法としては、ニューラルネットワーク、特に再帰型ニューラルネットワーク (Recurrent Neural Network : RNN) を用いる。

3. 研究の方法

(1) RTT (Round-trip Time) ごとの輻輳ウィンドウサイズの推定

提案方法では、通信ログから RTT ごとに輻輳ウィンドウサイズ $cwnd$ の推定を行う。パケットをキャプチャするモニタがネットワークの途中にあることを考慮して、以下のような方法を用いる。ここでは TCP タイムスタンプオプションが使用されていることを前提とする。

- 再送データに対する ACK セグメントなど、通信ログにおける特定の ACK セグメントに着目する。これを $ACK1$ と呼ぶ。
- 着目した ACK セグメント $ACK1$ の TSval (timestamp value) の値を TSecr (timestamp echo reply) に持つデータセグメントを探す。これを $DATA1$ と呼ぶ。
- データセグメント $DATA1$ を最初に確認応答する ACK セグメントを探す。これを $ACK2$ とする。さらに $ACK2$ の 1 つ前の ACK セグメントを $ACK1$ と呼ぶ。
- ACK セグメント $ACK2$ の TSval の値を TSecr に持つデータセグメントを探す。これを $DATA2$ と呼ぶ。

これらを用いて $ACK1$ を受信した時点での $cwnd$ を以下の式により推定する。

$$cwnd = \left\lfloor \frac{seq\ in\ DATA2 - ack\ in\ ACK1}{MSS} \right\rfloor \text{ (segments)}$$

ここで seq と ack は TCP ヘッダ中のシーケンス番号と応答確認番号を示す。MSS はその TCP コネクションで使用される最大セグメントサイズである。また記号 $\lfloor x \rfloor$ は x の整数部分を示す。

(2) 識別器への入力を選択と正規化

提案方法では輻輳回避フェーズにおける $cwnd$ の増加に着目する。そこでパケット損失の間 (ロスレス区間) の $cwnd$ の時系列を識別器の入力に用いることとする。しかし、ロスレス区間の長さはそれぞれ異なり、また $cwnd$ の値の範囲もそれぞれで異なる。そこで提案方法では、ロスレス区間の長い $cwnd$ 時系列を優先的に入力として選択し、さらに選択した $cwnd$ 時系列に対して時間軸と $cwnd$ 軸の双方で正規化することとした (サンプル数は 128)。

(3) RNN に基づく識別器の採用

提案方法では、TCP の輻輳制御アルゴリズムを出力とする識別器を構築するために RNN を採用した。ニューラルネットワークは様々なデータの分類・回帰・推定のために広く使用されているが、RNN は時間的な順序のある振る舞いを扱うために使用される。提案方法においても $cwnd$ の値の時系列を扱うため、RNN が適切であると判断した。さらに RNN の中でも長期的な依存関係を学習可能な LSTM (long short-term memory) を用いることとした。識別器への入力は上記のような正規化された $cwnd$ の時系列とし、輻輳制御アルゴリズムを one-hot ベクトルで表したものをラベルとす

表 1 RNN 識別器のハイパーパラメータ

Parameter	Value
Input Length	128
Hidden Layers	1
Hidden Neurons	512
Optimizer	Adam
Learning Rate	2×10^{-4}

る。RNN のハイパーパラメータを表 1 のように定める。隠れ層のニューロン数が 512 と、入力データ長 (128) に比べて大きく設定されている。これは隠れ層に冗長な表現力を持たせるためである。

(4) DRNN への拡張

続いて、隠れ層の段数を増やしたディープラーニングによる RNN (DRNN) に拡張する検討を行う。(3)と同様に LSTM を使用し、ハイパーパラメータとしては表 2 のような値を用いることとした。Optimizer などのパラメータについては複数の値からトレーニング時に決定するようにした。

表 2 DRNN 識別器のハイパーパラメータ

Parameter	Value
Input Length	128
Hidden Layers	3
Hidden Neurons	1st./2nd: 512, 3rd: 256
Optimizer	Adam or MomentumSGD
Learning Rate	$[10^{-5}, 10^{-1}]$
Weight Decay	$[10^{-10}, 10^{-3}]$

(5) 評価実験

評価実験については次の 2 つのステップで行う。

① ローカルなネットワークを用いた評価実験

実験室内に図 1 に示すネットワーク環境を構築する。送信側、受信側、ブリッジが 100Mbps イーサネットに接続され、ブリッジにおいて片方向それぞれ 50 ミリ秒の遅延を挿入した。その結果 TCP フローの RTT は 100 ミリ秒となる。また輻輳制御を起動するために、ブリッジにおいてランダムなパケット損失を挿入した。パケット損失率は 0.01% である。データ転送は iperf3 を送信側と受信側で動作させることにより行う。通信ログは送信側のイーサネットインタフェースにおいて、tcpdump を用いて収集した。

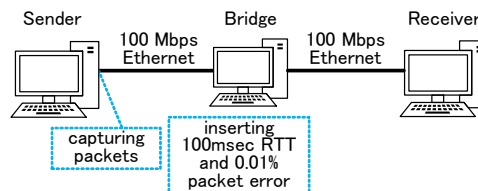


図 1 ローカルな実験環境

このネットワークを用いて、送信側の輻輳制御アルゴリズムを変更し、識別器のトレーニングを行う。対象とするアルゴリズムは、TCP Reno, HighSpeed TCP, Scalable TCP, BIC TCP, CUBIC TCP, Hamilton TCP, TCP Westwood+, TCP Vegas, TCP Veno, TCP BBR の 10 種類とする。トレーニングを行った識別器に対して、同じネットワークで収集したデータを用いてテストを行い、識別器の推定能力を判定する。

② インターネットに接続されたサーバの輻輳制御アルゴリズムの推定

ローカルなネットワーク環境で収集したデータによってトレーニングを行った識別器を用いて、インターネットに接続された実際のサーバの輻輳制御アルゴリズムを推定する実験を行う。トレーニングのためのローカル環境は図 1 と同様とする。ただし現実のサーバでの通信では RTT が異なってくるため、図 1 の環境において固定的な RTT を 100 ミリ秒、200 ミリ秒、500 ミリ秒に設定して、それぞれの場合のトレーニング用データを収集した。ここでも RTT の値は固定値であることに注意する必要がある。またパケット損失率は 0.01% のままである。

インターネットの代表的サーバと通信を行い、その通信ログからテストデータを作成する。このデータを識別器に適用し、そのサーバの輻輳制御アルゴリズムを推定する。対象とするサーバは Alexa Top Site List に示される 20,000 のサーバとする (なおこのサービスは 2022 年 5 月に終了した)。識別器が対象とする輻輳制御アルゴリズムは、上記の 10 種類に TCP Hybra, YeAH TCP, TCP Illinois を加えた 13 種類とする。なお Compound TCP は TCP Illinois に含むものとする。

4. 研究成果

(1) RNN ベース識別器による輻輳制御アルゴリズムの推定

TCP 輻輳制御アルゴリズムを推定するための RNN 識別器を Google から提供されている TensorFlow ライブラリを用いて実装した。通信ログの取得、正規化、識別器のトレーニング用データとテスト用の取得、識別器のトレーニングとテストによる輻輳制御アルゴリズムの推定について次のようなステップを用いた。

- 各輻輳制御アルゴリズムに対してトレーニング用データとテスト用データを得るために、iperf3 を用いた 10 分間のデータ転送を 10 回ずつ、計 20 回行った。1 回ずつの結果に対して、通信の開始時点以外のロスレス区間の正規化された cwnd 時系列を求め、サンプル数の多い順に並べた。これらの結果から、トレーニング用およびテスト用に、サンプル数の多いものから 400 ずつの cwnd 時系列を選択し、識別器への入力とした。この処理は輻輳制御アルゴリズムごとに行った。
- 得られた入力に基づいて、トレーニングとテストを繰り返して行った。トレーニングについては、1 回のトライアルに対して、ミニバッチサイズ分の入力を 400 の cwnd 時系列からランダムに選び識別器に与え、識別機の出力と入力のラベルとが一致するかを調べた。これを対象となるすべての輻輳制御アルゴリズムに対して行い、アルゴリズムごとの識別精度と全体の識別精度を求めた。
- 1 回の学習トライアルが終わると、テストトライアルを実行した。テストトライアルでは、輻輳制御アルゴリズムごとに 400 の入力をすべて用いた、その結果からテスト用データに対する推定の識別精度 (アルゴリズム別と全体) を計算した。
- これらの実験を繰り返して行い、トライアルの回数 (エポック数) に対する識別精度の変化を求め学習過程を評価した。

図2にこの実験における混同行列を示す。10の輻輳制御アルゴリズムの内、BIC TCP、CUBIC TCP、Hamilton TCP、TCP BBRについては有効に識別することができている。一方HighSpeed TCPとScalable TCPは混同して推測されてしまっている。また、TCP Reno、TCP Westwood+、TCP Vegas、TCP Venoについては互いに混同して識別されている。これはこれらのアルゴリズムがTCP Renoと同様なAIMD(Additive Increase Multiplicative Decrease)に従っているためであると考えられる。

(2) DRNN ベース識別器による輻輳制御アルゴリズムの推定

続いて、DRNN 識別器を用いた輻輳制御アルゴリズムの推定を行った。識別器のハイパーパ

表3 使用したDRNN 識別器のハイパーパラメータ

Parameter	Value
Optimizer	Adam
Learning Rate	0.0015967736
Weight Decay	2.987486×10^{-6}

ラメータは、表3に示すような値に最適化した。

図3にこの識別器による10のアルゴリズムの推定の混同行列を示す。RNN ベースの識別器の場合に比べて、推定精度が向上していることがわかる。全体の推定精度は80%程度となった。高い精度の推定が可能となったといえる。

(3) AlexaTop20000 サーバの輻輳制御アルゴリズムの推定

上記のようにインターネットに接続された代表的なサーバの輻輳制御アルゴリズムの推定を行った。Mishraらはテスト系列を用いてサーバの輻輳制御アルゴリズムの推定を行った[1]。それによると上述の13のアルゴリズムが使用されていると報告されているため、新たに3つのアルゴリズムを追加した。その結果(混同行列)を図4に示す。全体の推定精度は72%程度となった。

この識別器を用いてAlexa Top Listの20000のサーバに対してアルゴリズムを推定した。その分布を図5に示す。この結果は文献[1]の結果と同様な傾向を示しており、作成した識別器は有効であると考えられる。

<引用文献>

[1] A. Mishra, et al., "The Great Internet TCP Congestion Control Census," Proc. ACM Meas. Anal. Comput. Syst., vol. 3, no. 3, article 45, pp. 1-24, 2019.

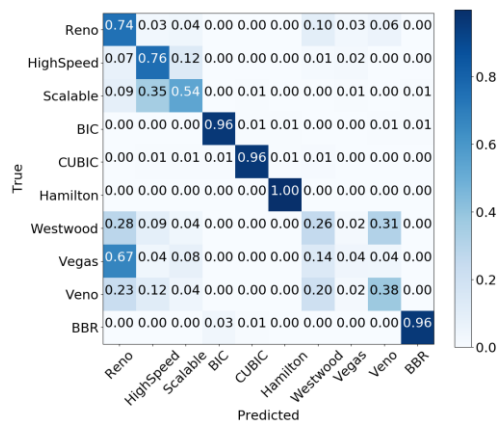


図2 RNN 識別器による10のアルゴリズムに対する混同行列

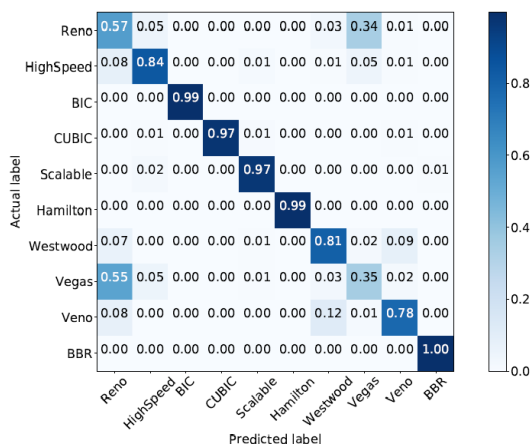


図3 DRNN 識別器による10のアルゴリズムに対する混同行列

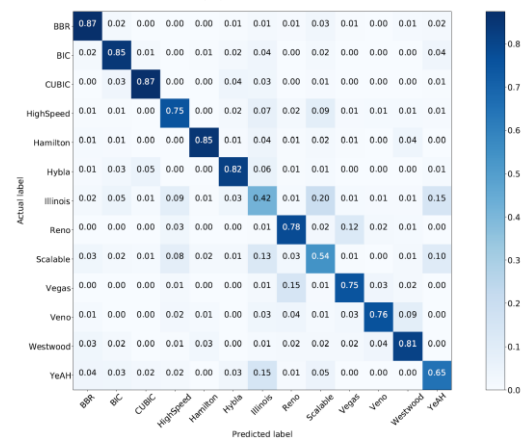


図4 13のアルゴリズムに対する混同行列

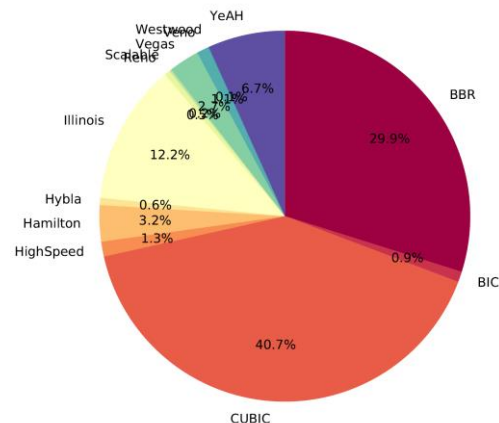


図5 AlexaTop20000サーバに対する推定結果

5. 主な発表論文等

〔雑誌論文〕 計2件（うち査読付論文 2件 / うち国際共著 0件 / うちオープンアクセス 1件）

1. 著者名 大関 尚紀、山本 嶺、大坐島 智、加藤 聡彦	4. 巻 Vol. 61, No. 2
2. 論文標題 リカレントニューラルネットワークを用いた双方向通信ログに基づくTCP輻輳制御の推定方式	5. 発行年 2020年
3. 雑誌名 情報処理学会 論文誌 (ジャーナル)	6. 最初と最後の頁 287-296
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 -

1. 著者名 Tetsuya Sawada, Ryo Yamamoto, Satoshi Ohzahata, Toshihiko Kato	4. 巻 No. 1&2
2. 論文標題 TCP Congestion Control Algorithm Estimation by Deep Recurrent Neural Network and Its Application to Web Servers on Internet	5. 発行年 2023年
3. 雑誌名 International Journal On Advances in Networks and Services	6. 最初と最後の頁 1-9
掲載論文のDOI (デジタルオブジェクト識別子) なし	査読の有無 有
オープンアクセス オープンアクセスとしている (また、その予定である)	国際共著 -

〔学会発表〕 計2件（うち招待講演 0件 / うち国際学会 2件）

1. 発表者名 Naoki Ohzeki, Ryo Yamamoto, Satoshi Ohzahata, Toshihiko Kato
2. 発表標題 Estimating TCP Congestion Control Algorithms from Passively Collected Packet Traces using Recurrent Neural Network
3. 学会等名 10th International Conference on Data Communication Networking (国際学会)
4. 発表年 2019年

1. 発表者名 Tetsuya Sawada, Ryo Yamamoto, Satoshi Ohzahata, Toshihiko Kato
2. 発表標題 Estimation of TCP Congestion Control Algorithms by Deep Recurrent Neural Network
3. 学会等名 14th International Conference on Emerging Networks and Systems Intelligence (EMERGING 2022) (国際学会)
4. 発表年 2022年

〔図書〕 計0件

〔産業財産権〕

〔その他〕

-

6. 研究組織

	氏名 (ローマ字氏名) (研究者番号)	所属研究機関・部局・職 (機関番号)	備考
--	---------------------------	-----------------------	----

7. 科研費を使用して開催した国際研究集会

〔国際研究集会〕 計0件

8. 本研究に関連して実施した国際共同研究の実施状況

共同研究相手国	相手方研究機関
---------	---------