

深層学習を用いた複数端末の無線 LAN 通信時のパケット解析と輻輳の予測

理学専攻・情報科学コース 1840675 山本 葵 (指導教員：小口正人)

1 はじめに

近年、世界中に増え続けているワイヤレスデバイスは機能や性能も強化され、気軽にネットワークにアクセスすることができ、大容量かつ高速な通信に対する需要は増大している。しかし有線接続に比べ低帯域かつノイズの多い無線接続においては、膨大なパケットが通信中に無線 LAN アクセスポイントに蓄積され、輻輳が発生してしまうという問題も生じている。本研究では輻輳発生前に制御を加え無線 LAN AP の輻輳を回避することを最終目的とし、本稿では目的達成のため輻輳の予測を行う。端末を用いて無線 LAN 通信を行い、アクセスポイント (AP) 周りのパケットを深層学習の LSTM モデルを用いて解析し無線 LAN 通信時のトラフィックの予測性能を評価した。

2 関連研究

2.1 カーネルモニタ

先行研究 [1] で開発されたカーネルモニタは、通常見ることができない輻輳ウィンドウ (CWND 値) や RTT などの様々な TCP パラメータをモニタできるツールである。このツールを Android に組み込むことで [2][3], TCP 通信時におけるカーネル内部の処理を、ユーザ空間からリアルタイムに解析し、各パラメータ値の変化を記録する。

本研究はカーネルモニタで得ることができる情報を用いて、無線 LAN AP に接続される端末において輻輳を回避するために制御を行うことを目標としている点で関連研究を継承しているが、輻輳が起こった後に制御を加える関連研究に対し、本研究は輻輳が起こることを予想して輻輳が起こる前に制御を加えることを目標としている点で異なる。

3 深層学習

本研究では深層学習を行う。深層学習はニューラルネットワークの階層を深めたアルゴリズムで機械学習を実装するための 1 つの手法である。機械が自分自身で特徴量を抽出、階層を深めることで精度が大幅に向上した。LSTM(Long Short-Term Memory) は時系列データに対するモデルである。LSTM は隠れ層のユニットを LSTM block と呼ばれるメモリと 3 つのゲートをもつブロックにすることで実現された。その最も大きな特徴は文章などの長い依存関係にあるデータも覚えておくことができる長期依存が可能であるということである。本研究では時系列データであるパケットの解析であるため、この LSTM をモデルとした深層学習を行う。

4 無線 LAN 通信における TCP パラメータと通信速度のふるまい

本章では深層学習をする際の正解データをどのように設定するか検討する。輻輳を示す値としてどのパラメータを採用すべきか実験を行い決定する。先行研究にて開発され

たカーネルモニタを導入した Android 端末を用いて通信を行い、TCP パラメータを取得する。また、通信には iperf を使用して通信速度であるを測定する。輻輳の発生を観察するために 3600 秒間パラメータを取得し、10 秒付近で全ての Android 端末から一斉にパケットを送信した。

4.1 0-200 秒間の合計通信速度と CWND 値

図 1 は Android 端末 5 台で通信をおこなったときの 0-200 秒での合計通信速度と cwnd 値の振る舞いである。10 秒で一斉にパケットを送信したときは大きく cwnd 値があがっているが、その後は多少の値の増減はあるが、5 台の端末はほぼ同じような値を維持している。また合計通信速度も急激な変化はなく、5 台の端末が公平に通信を行なっていることがわかる。

4.2 1770-2010 秒間の合計通信速度と CWND 値

図 2 は 3600 秒測定しているなかの 1770-2010 秒での合計通信速度と cwnd 値の振る舞いである。1790 秒付近で cwnd 値が大きく下がったあと、5 台ともほぼ同じ値を維持していた cwnd 値の均衡が崩れ、一部の端末のみが帯域を占有している様子が見られる。cwnd 値の大きな変化に伴い、合計通信速度も大きく増減し始めている様子が見られる。

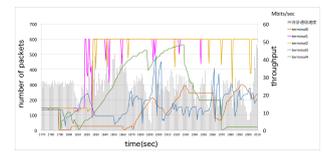
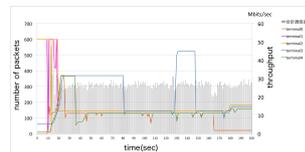


図 1: 0-200 秒の合計通信速度と CWND 値の推移 図 2: 1770-2010 秒の合計通信速度と CWND 値の推移

4.3 輻輳を示す値 (予測する値)

実験により輻輳を示す値として合計通信速度を用いる。合計通信速度が急激に減少するときは AP が非常に混雑していると考えられ、また通信をしたいのにできていない状態というのはネットワークが正常ではないと考えられる。一部の端末のみが帯域を占有し、各端末が十分な通信速度を出すことができない状態は、今後制御することで避けるべき状態であるので合計通信速度を深層学習の正解データとして予測する。しかし、測定した合計通信速度は瞬間的な増減が多くみられ、全体的な合計通信速度の増減をとらえない本研究ではノイズとなってしまう可能性がある。そのためある程度合計通信速度をなだらかにし、ノイズを抑えるため $t-4$ から t 秒までの移動平均をとる。移動平均を取る前の移動平均と移動平均を取った後の移動平均を図 3 に示す。移動平均を用いることで急減急増が抑えられている。よって次の章では移動平均をとった合計通信速度を予測できるか実験する。

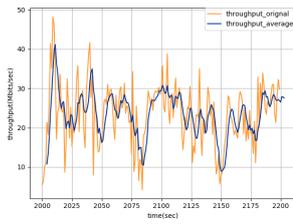


図 3: 移動平均を用いた合計通信速度

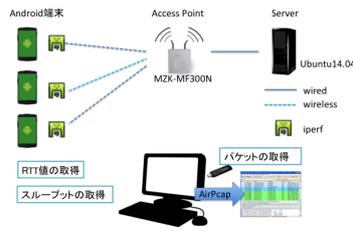


図 4: 実験環境

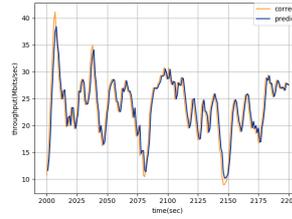


図 5: 実験 1 学習データによる予測結果

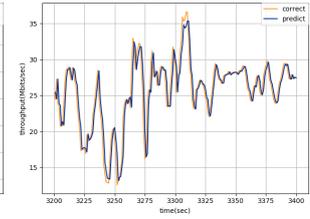


図 6: 実験 1 テストデータによる予測結果

5 予測実験

本章では Android 端末によるデータ通信を行なった時に深層学習を用いて輻輳が予測できるか評価実験を行なった。今回の実験に用いた深層学習用フレームワークは Chaier である

5.1 実験環境

4 章のデータを取得し実験を行う。実験環境を図 4 に示した。1 台の AP に接続した複数台の Android 端末から iperf によって、データをサーバに送信する。それと同時に PC で上記のデータを取得する。その後、LSTM モデルを用いた深層学習をおこなう。

データセットは表 1 である。移動平均を用いた合計通信速度を使用するため、 $t-4$ 秒から t 秒までの移動平均を取った合計通信速度から $t-3$ 秒から $t+1$ 秒の合計通信速度を予測するよう学習を行なった。4 章でも使用した Android 端末 5 台で 3600 秒通信したデータを使用する。今回は一部の端末が帯域を占有し、合計通信速度が大きく減少する箇所を予測できるか検証するため 1800-3000 秒のデータを使いデータセットとし学習を行う。

表 1: 実験 1 データセット

入力データ		正解データ
$t-4$ から t まで 移動平均	合計通信速度	合計通信速度

5.3 予測結果

図 5 が 1800-3000 秒の学習用データを学習させたモデルに再び同じ学習用データを入力データとして与えたものである。図は 2000-2200 秒を抜き出した結果である。図 6 は学習で使ったデータの後 3000-3400 秒をテスト用データとして学習済みのモデルに入力した結果である。オレンジの線が正解の合計通信速度、青の線が予測した値である。

精度よく学習できているように見えるが、1 秒前の入力結果を出力するように学習されているようにみえる。

5.4 誤差評価

提案手法による予測は 1 秒前の入力結果を出力するように学習しているように見える。一つ前の入力をそのまま出力する repeat 手法と誤差を比較することで評価を行う。テストデータによる予測結果と式 8.1 を用いて N 乗誤差平均を算出する。算出した結果を図 7 に示す。提案手法による予測の方が誤差が少ないことがわかり、repeat 手法より提案手法の方が予測が優れていることがわかる。さらに N を 2 から 3 に増加させると、予測誤差と repeat 誤差の差が広がる。これは提案手法による予測結果の方が repeat 手法に

よる予測結果と比較して大きな予測はずしをしていないということである。

$$N \text{ 乗誤差平均} = \text{average}(|\text{correct}_t - \text{predict}_t|^N) \quad (1)$$

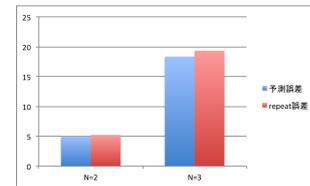


図 7: 提案手法と repeat 手法による誤差の比較

6 おわりに

本研究では、AP に接続する端末が複数台通信を行い、輻輳発生前に制御を行うことを目標とし、その前段階として輻輳の極めて早期の検出、予兆の発見をするために深層学習を用いてトラフィックの予測を行なった。また深層学習による予測が repeat 手法より優れていることを証明した

今後の課題としては様々な状況下でデータ通信を行なっている時の予測精度の向上が 1 番に挙げられる。数時間、数日とさらに長時間のデータを集めてあらゆる状況においてのデータを用いて十分に学習をさせていきたい。

また今後輻輳制御を行っていくことを見据え、計算時間も考慮に入れて実験を行っていききたい。どのくらい前に輻輳が発見できれば制御に間に合うのか明らかにし、その時間を考慮して予測実験を今後行っていききたい。

参考文献

- [1] Kaori Miki, Saneyasu Yamaguchi, and Masato Oguchi: "Kernel Monitor of Transport Layer Developed for Android Working on Mobile Phone Terminals," Proc. ICN2011, pp.297-302, January 2011.
- [2] Ai Hayakawa, Saneyasu Yamaguchi, Masato Oguchi: "Reducing the TCP ACK Packet Backlog at the WLAN Access Point," Proc. ACM IMCOM2015, 5-4, January 2015.
- [3] Ayumi Shimada and Masato Oguchi: "A Study of Android Tables Performance," Proc.DEIM2017,H2-3,March 2017