

環境情報を利用した上位層における通信制御手法

飯尾 明日香 (指導教員：小口 正人)

1 はじめに

現在、自動車には数百種類のセンサが搭載されており、自動車はこれらのセンサから車両間隔、スピード、周辺端末数、加速度など数多くの情報を取得できる。このようなセンサから得た周囲の環境情報は、自動車において言えば、事故防止・車両安定制御等に役立てられている。

一方、近年 Wireless Local Area Network(WLAN)による通信が一般化しているが、WLANにおいて通信における各パラメータは静的に設定されているため、その場の環境に応じた通信設定は行われない。したがって、センサから環境情報を取得可能な端末であっても資源を効率的に利用した通信が行われておらず、必ずしも最適ではない状態で通信を行っているという問題がある [1][2]。そこで、本研究では、環境情報を適切な無線通信パラメータ設定に利用し、各端末が周囲の状況に応じた設定を行うことで、通信効率を向上させる手法の検討を行う。

本稿では、環境情報の利用先として上位層の Transmission Control Protocol(TCP)に着目した。TCPにおいて、確認応答(ACK)を受けずに送信できるパケットの最大数である輻輳ウィンドウ(cwnd)サイズは輻輳制御アルゴリズムにより制御されているため、各端末は必ずしも周辺の状況に合った通信を行っているとは言えない。そこで、輻輳制御アルゴリズムによりcwndサイズが制御される一般的なTCPを用いて通信を行った場合に対し、cwndサイズを一定値とするTCPを用いて通信を行った場合において、端末数や往復遅延時間(RTT)に応じたスループットの変化をシミュレーションにより測定し、比較する。

なお、実験にはネットワークシミュレータ OM-NeT++ version4.1を使用した。

2 TCP

2.1 TCPによる輻輳制御

WLANの国際標準としてIEEE 802.11が規定されており、IEEE 802.11ではTCPにより輻輳制御が実現されている。輻輳とはネットワークの混雑を表す言葉であり、ネットワーク上で多量のトラフィックが発生して、通常の通信が困難になることをいう。TCPは、輻輳制御により輻輳を回避する仕様となっている。代表的なTCPとしてTahoe, Reno, NewReno, Vegas等があり、輻輳制御アルゴリズムはTCPによって様々である。

TCPでは、ウィンドウフロー制御をすることによって、輻輳の防止・早期回避を可能にしているが、一方で、輻輳回避フェーズにおける輻輳によるcwndサイズの減少幅が、後の線形増加時の増加幅と比較し大きいいため、一度輻輳が起きcwndサイズが下がると、cwndサイズが増加するまでに時間がかかり、輻輳制御が原因となったスループットの低下が起こる場合がある。以下では、本稿で用いたTCPであるTCP TahoeとTCP Renoの輻輳制御アルゴリズムについて説明する。

2.2 TCP Tahoe

TCP Tahoeは、スロースタートフェーズと呼ばれる、ACKセグメントを受信した分だけcwndサイズを増加させていく動作を行う(式(1), if条件式)。

$$cwnd \leftarrow \begin{cases} cwnd + 1 & \text{if } cwnd < ssthresh \\ cwnd + \frac{1}{cwnd} & \text{otherwise} \end{cases} \quad (1)$$

ここで、ssthreshはスロースタートから輻輳回避のフェーズへ移行する際の閾値とする。ssthresh初期値は多くの実装で広告ウィンドウサイズが設定されるが、ある程度大きい値であれば任意である。また、輻輳が起こるとssthreshは以下のように更新される。

$$ssthresh \leftarrow \frac{ssthresh}{2} \quad (2)$$

その後、cwndサイズを1まで下げ、再びスロースタートフェーズを行う。やがてcwndサイズがssthreshに達すると輻輳回避フェーズへ移行し、cwndサイズを式(1)otherwise条件式を用いて線形的に増加させる。ここで再度輻輳が起こると、再び式(2)を用いてssthreshを更新し、cwndサイズを1まで下げ、スロースタートフェーズへ移行する。TCP Tahoeはこの動作を繰り返す。

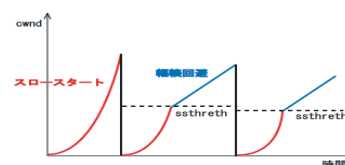


図 1: TCP Tahoe における cwnd の変化

2.3 TCP Reno

TCP RenoはTCP Tahoeがベースのアルゴリズムで、TCP Tahoeに対してエラーが起きても必ずしもcwndサイズを1まで下げないように改良したものである。すなわち、始めはスロースタートフェーズでcwndサイズの急速な増加を図り、その後、cwndサイズがssthreshに到達すると、輻輳回避フェーズへと移行する。輻輳回避フェーズでは、輻輳が起こるとssthreshを半分に更新し、cwndサイズをssthreshまで下げたから線形的に増加させ、輻輳が起こる度にこれを繰り返す(図2)。

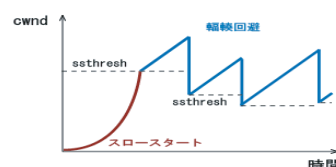


図 2: TCP Reno における cwnd の変化

3 検証実験

3.1 概要

本稿では、複数の端末(クライアント)と1台のサーバが通信を行っており、クライアントから送信されたパケットが、サーバ側でSink(パケット破棄)されるシナリオを想定した(図3)。このとき各クライアントは、WLANでアクセスポイント(AP)へ、APから有線でルータを2台経由しサーバへパケットを送信している。

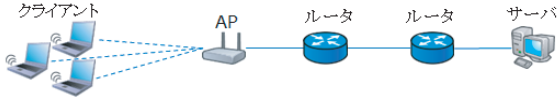


図 3: 実験に用いたモデル

ここで、輻輳制御が行われている TCP(TCP Tahoe, TCP Reno) と、輻輳制御が行われない TCP(No Congestion Control) の3つの輻輳制御モデルを用い、それぞれのクライアント数に応じたスループットを測定した。TCP Tahoe と TCP Reno において、cwnd サイズは式 (1) および (2) が適用され動的に更新されながら通信が行われる。一方、No Congestion Control における cwnd サイズは輻輳が起きても下げられず一定値を保つ設定で通信が行われる。このシナリオでは、無線規格として IEEE 802.11 を使用し、その他のパラメータは表 1 のように設定しシミュレーションを行った。結果を図 4, 図 5 に示す。

なお、図 4 の実験の際、RTT は 100msec に固定している。また、複数台での通信を想定しているためクライアント数は 2~50 とした。

表 1: シミュレーションパラメータ

Send Bytes	2MByte
Advertised Window	8,355,840Byte
Data Rate	6Mbps
Carrier Frequency	2.4GHz
Maximum Segment Size	65,280Byte
Maximum Sending Power	20mW

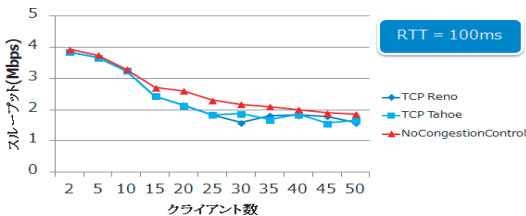


図 4: 端末数に伴うスループット

3.2 考察

図 4 から、RTT が 100ms の場合では NoCongestionControl がクライアント数に関わらず最も通信性能が良いことがわかる。これらの性能差から、クライアント数が大きい場合に、TCP Tahoe や TCP Reno の輻

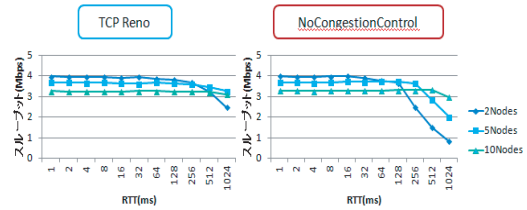


図 5: RTT に伴うスループットの比較

輻輳制御が必ずしも効率良く機能しておらず、結果的に、cwnd サイズに制限をかけない NoCongestionControl の性能が高くなったものと考えられる。

次に、RTT に伴うスループットの変化を測定したところ、クライアント数が小さいほど、RTT の増加に伴いスループットが大きく低下する様子が見られた(図 5)。クライアント数の大きい場合、高遅延で必要となる大きな cwnd サイズを分担できるためスループットが低下しにくくなることから、これは妥当な結果であると言える。両者を比較すると、NoCongestionControl のスループットがより大きく低下している。これは、NoCongestionControl では cwnd サイズをある値に固定しており、遅延が小さい場合にはそれが十分なサイズであるが、遅延が大きい場合にこの設定が適切ではなく、通信を制限してしまったため、TCP Reno の性能を下回ったと考えられる。

4 まとめと今後の課題

WLAN において遅延環境モデルを用い、一般に使用されている輻輳制御を行う TCP Tahoe および TCP Reno と、輻輳制御を行わない TCP のそれぞれの端末数に伴うスループットを測定し、従来の TCP では端末数に応じた最適な通信が行われていない可能性があることがわかった。また、RTT に伴うスループットを比較し、高遅延環境では TCP Reno よりも cwnd サイズを一定に保つ TCP の性能が大きく落ちた。これらの結果から、この cwnd サイズの設定方法を改変することで高遅延環境における通信性能を向上できると考えられるため、今後は、cwnd サイズを任意の値に固定できるような TCP を実装し、cwnd サイズに伴うスループットを測定することで、周辺の状況に応じた最適な通信設定の手法を検討したい。

参考文献

- [1] Joseph Camp, Onur Altintas, Rama Vuyyuru, and Dinesh Rajan, "Context-aware Cpllection, Decision, and Distribution (C2D2) Engine for Multi-Dimensional Adaptation in Vehicular Networks", VANET '11, pp.85-86, September 23, 2011
- [2] 松本真紀子, Onur Altintas, 西堀満洋, 小口正人: "環境情報を利用したマルチプルアクセス手法の一検討", 情報処理学会第 73 回全国大会, 1V-6, March 2011
- [3] 飯尾明日香, 小口正人: "無線環境における環境情報を利用した上位層における通信制御手法の検討", DEIM2012, 2012 年 3 月 発表予定