

インターネットに接続された無線アドホックネットワークを指向した 受信側 TCP 主導の送信レート制御に関する一検討

小嶋 明寿^{†1} 石原 進^{†2}

一般に無線アドホックネットワークでは TCP の通信速度が低くなってしまうことが知られており、TCP 通信の性能を向上させる手法が多く提案されている。これらの手法の多くは送信側 TCP がアドホックネットワークに対応していることを前提としている。しかし、無線アドホックネットワークをインターネットに接続し、アドホックネットワーク側から Web 等を利用する場合、多くの送信側 TCP はインターネット上のサーバにある。そのため、送信側 TCP がアドホックネットワークに対応していることは期待できない。従って、このような場合、送信レート制御を受信側の TCP が行う必要がある。筆者らは受信側主導の輻輳制御手法として RSC-VAR (Receiver initiated Sending-rate Control based on TCP-VAR) を提案している。しかし、RSC-VAR は無線アドホックネットワークを有線ネットワークに接続したとき、有線ネットワーク側の遅延が大きいと高いスループットを得ることができないことがわかった。これは、RSC-VAR が送信レート決定に End-to-End での SmoothedRTT を利用しているためである。本稿では RSC-VAR の送信レート決定手法を改良し、SmoothedRTT に依存しない送信レート決定手法を提案する。シミュレーションにより改良手法を評価した結果、改良以前に対し約 4 倍のスループットを得ることができた。

A Study of Receiver Initiated Sending Rate Control based Data Receive Rate on Ad Hoc Networks connected to the Internet

AKIHISA KOJIMA^{†1} and SUSUMU ISHIHARA^{†2}

Due to the low performance of TCP on ad hoc networks, many techniques for improving the performance on ad hoc networks have been proposed. Most of them change the function of the TCP sender. When an ad hoc network is connected to the Internet and the user in the ad hoc networks uses web, most of the sender TCPs on servers are on the Internet. TCP on such servers are not adapt to ad hoc networks. In this case, the sending rate should be controlled by the receiver TCP. We have proposed a Receiver initiated Sending-rate Control based on TCP-VAR (RSC-VAR) scheme, in which a receiver calculates the sending rate optimized for ad hoc networks using algorithm used in TCP-VAR and controls the sending interval of ACKs and its advertisement window. However, RSC-VAR can not achieve high throughput when the ad hoc network is connected to the Internet and the delay of the route on the Internet is long, because RSC-VAR calculates sending rate using End-to-End smoothed RTT. In this paper, we propose an improved scheme for calculating the sending rate without using smoothed RTT. Simulation results show that the proposed scheme achieves four times as high throughput as the original RSC-VAR.

1. はじめに

無線アドホックネットワークにおいて TCP では高いスループットを得ることが困難であることが知られている¹⁾²⁾。この要因の 1 つに自己干渉の影響がある。無線アドホックネットワークでは、同一経路に連続的にパケットが送信されると、同一フロー上の中継端末間で無線通信経路での競合を起すため、スループットが低下する。この問題の解決策として、パケットの

送信間隔を適切に空ける送信間隔制御手法 TCP-AP³⁾、TCP-VAR⁴⁾ 等が提案されている。

しかし、これらの手法は送信受信双方の TCP、もしくは送信側の TCP を変更する必要がある。そのため、無線アドホックネットワーク内からゲートウェイを介してインターネットに接続して Web 等のサービスを利用する場合、これらの手法の応用は困難である。例えば、無線アドホックネットワーク側からの Web アクセスを想定すると、ほとんどの場合、送信側 TCP はインターネット上のサーバにある。これらインターネット上の全てのサーバが TCP-AP、TCP-VAR 等に対応していることを期待することは現実的ではない。

^{†1} 静岡大学大学院工学研究科
Graduate School of Engineering, Shizuoka University

^{†2} 静岡大学創造科学技術大学院
Graduate School of Science and Technology, Shizuoka University

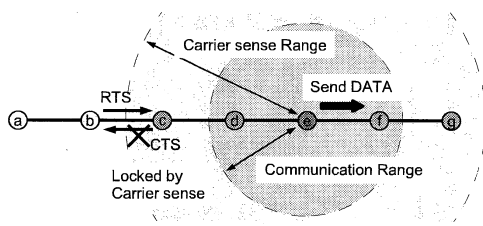


図1 直線型トポロジでの干渉

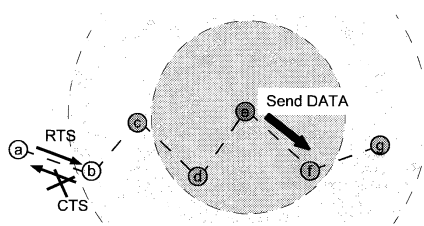


図2 ジグザクなトポロジでの干渉

筆者らは、無線アドホックネットワークにおいて受信側 TCP が送信側 TCP のデータセグメント送信を制御する送信間隔制御手法 RSC-VAR (Receiver initiated Sending-rate Control based on TCP-VAR) を提案している⁵⁾⁶⁾。しかし、RSC-VAR は有線ネットワークにつないだとき、有線側の遅延が大きいと高いスループットを得ることができないという問題がある。そこで、本稿ではその問題点を明確にし、RSC-VAR の送信レート決定のための改良アルゴリズムを提案する。RSC-VAR では送信レートの算出に SmoothedRTT(sRTT) を用いていたため、有線側の遅延が大きい場合、送信レート決定アルゴリズムが有効に働かなかった。改良アルゴリズムでは送信レートを sRTT を用いず、実効スループットの揺らぎに基づいて制御する。

以降、本稿では、2章で受信側主導の送信レート制御手法 RSC-VAR について述べ、3章で RSC-VAR の改良手法を提案する。4章ではこの改良手法をシミュレーションにより評価し、最後に5章で本稿をまとめる。

2. RSC-VAR (Receiver initiated Sending-rate Control based on TCP-VAR)

2.1 無線アドホックネットワークにおける TCP の送信間隔制御

無線アドホックネットワークでは特定の場所にトラフィックが集中すると近隣のノード間で競合を起し、ネットワークの性能が著しく低下してしまうという特徴がある。TCP は送信ウィンドウに空きがある場合、同一経路上に連続的にパケットを送信する。このため、無線マルチホップのネットワークで TCP 通信を行うと、同一フローのパケット同士が互いに干渉し合い、リンクの利用効率を低下させてしまう。例えば、図1において、端末 a から端末 g に向けて TCP 通信が行われており、今、端末 e が端末 f に向けてパケット送信中であるとする。このとき、端末 c、d はキャリアセンスをしているため、通信を行えない。また、端末 b も端末 c がキャリアセンス中であるため、端末 c に RTS

を送っても CTS が受信できず、端末 c にパケットを送信することができない。この自己干渉の問題を回避するため、ElRakabawa らはデータ送信間隔を適切に空ける TCP-AP を提案した³⁾。TCP-AP では先に送信されたパケットが 4 ホップ先に到達してから次のパケットを送信するように送信側でのセグメント送信間隔を調整する。文献 3) では、適切な送信間隔を Four-Hop propagation Delay (FHD) であるとしている。しかし、トポロジが複雑であったり、端末間の距離が異なっている場合、必ずしも FHD が適切な送信間隔であるとは限らない。例えば、図2では、自己干渉を回避するためには 5 ホップ分送信間隔を空ける必要がある。また、すべての中継端末の通信速度やバックグラウンドトラフィックの影響を考慮した FHD を正しく計算するのは困難である。これらの問題を解決するため、Chen らは適切な送信レートを予測する手法として TCP-VAR を提案した⁹⁾。TCP-VAR では受信した ACK セグメントから実効スループットを計算し、スループットの揺らぎを基に送信レートを決定する。

しかし、TCP-AP、TCP-VAR では送信側で送信レート制御を行うため、送信側 TCP がこれらの手法に対応している必要がある。したがって、送信側がインターネット上の一般的な TCP である場合には利用できない。そこで、筆者らは受信側で送信側のデータ送信を制御する手法 RSC-VAR を提案している。

2.2 RSC-VAR

RSC-VAR は、受信側で適切な送信レートを予測し、予測送信レートに基づき ACK 送信間隔を調節することで送信側のデータセグメント送信間隔を制御する手法である。TCP はデータ送信制御にスライディングウィンドウ方式を採用している。そのため、送信側は ACK セグメントを受信すると、その ACK セグメントに従い送信ウィンドウをスライドさせ、新たなデータセグメントを送信する。つまり、データセグメントの送信は ACK セグメントの受信に依存する。したがって、受信側 TCP が適当な ACK 番号を持ち、かつ適当な大きさの広告ウィンドウがある ACK セグメント

を適当な間隔で送信することで受信側主導で送信側のデータ送信を制御することが可能となる。

RSC-VAR の受信側ではデータセグメントを受信すると TCP-VAR と同様のアルゴリズムに従い送信レートを計算し、受信したデータセグメントに対する ACK セグメントを生成して、送信待ちキュー (ACK キュー) に追加する (図 3)。

ACK キューに入れられた ACK セグメントは、計算された送信間隔で ACK キューの先頭から取り出され、広告ウィンドウ値、タイムスタンプ値を挿入後、送信される。送信側は、この ACK の到着を契機として、新たにデータセグメントを 1 つ送信する。

送信レート予測

RSC-VAR では、TCP-VAR と同様、実効スループットの揺らぎからネットワークの輻輳の度合いを推測し、適切な送信レートを予測する。一般にネットワークが混雑していない場合、スループットの揺らぎは小さくなり、輻輳状態では大きくなる。したがって、RSC-VAR ではスループットの揺らぎが小さい状態が続くと送信レートを徐々に上げ、揺らぎが大きい状態が続くと送信レートを下げる。

実効スループット (AT : Achieved Throughput) の揺らぎはデータセグメントを受信するごとに計算される (式 (1)~(3))。

$$AT = (\text{received data in last } srttt) / srttt \quad (1)$$

$$AVE_{AT} = \frac{\sum_{i=1}^K AT_i}{K} \quad (2)$$

$$VAR_{AT} = \frac{\sum_{i=1}^K |AT_i - AVE_{AT}|}{(K-1) \cdot AVE_{AT}} \quad (3)$$

ここで、 $srttt$ は $sRRTT$ 、 AVE_{AT} 、 VAR_{AT} はそれぞれ AT の平均、揺らぎを表し、 K は AVE_{AT} 、 VAR_{AT} の計算に用いる AT のサンプル数を表す定数である。また、 K は VAR_{AT} が閾値 θ_{upper} 以上の場合とそれ以外の場合で値を変更する ($K_{congestion}$ 、 K_{normal})。これにより、ネットワークが輻輳状態である場合は局所的なスループットの変動を、ネットワークが混んでいない場合は長期的なスループットの変動を評価する。

RSC-VAR は送信レート (R)、および ACK 送信間隔 (T_{ack}) を、 VAR_{AT} 、 $srttt$ 、制御変数である仮想輻輳ウィンドウ ($vcwnd$) を用い以下の式 (4)、(5) にしたがって決定する。

$$R = \frac{vcwnd / srttt}{1 + \exp(VAR_{AT})} \quad (4)$$

$$T_{ack} = \frac{1}{R} = \frac{srttt}{vcwnd} \cdot (1 + \exp(VAR_{AT})) \quad (5)$$

制御変数 $vcwnd$ は VAR_{AT} が連続で N 回以上、閾値

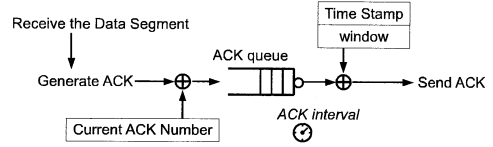


図 3 Adaptive ACK Pacing

(θ_{lower}) 以下のとき、定数 α' だけ線形に増加する。また、 VAR_{AT} が M 回以上連続で閾値 (θ_{upper}) 以上であれば、 $vcwnd$ は定数 β' ($0 < \beta' < 1$) 倍される。 $vcwnd$ の初期値は 1 である。通信開始から $vcwnd$ が初めて減少するまでをスロースタート状態とし、この期間は N を通常より小さな値 (N_{slow}) に設定する。パケットロスが発生すると $vcwnd$ は半減される。

Adaptive ACK Pacing

Adaptive ACK Pacing は受信側 TCP において送信レートに基づいた送信間隔で ACK セグメントを送信するための手法である。この手法では生成された ACK セグメントを一旦、バッファ (ACK キュー) に格納し、その後、送信レートによって決定される適切なタイミングで ACK キューの先頭の ACK を 1 つずつ取り出し、送信する (図 3)。

ただし、タイムスタンプオプションのような送信時に付加されるべき情報に関しては、ACK 送信時に最新のものを付加する。また、ACK 送信タイミングに ACK キューが空のときには、ACK を送信しない。

広告ウィンドウ値の計算

RSC-VAR では送信側のデータ送信間隔が、受信側の ACK 送信間隔に同期するように、受信側が広告ウィンドウの大きさを調整する。一般に、受信側 TCP の受信バッファの空き容量が十分にある場合、受信側 TCP は大きな広告ウィンドウを送信側に通知する。この場合、送信側の送信ウィンドウは送信側の持つ輻輳ウィンドウによって制限されることになる。送信側の輻輳ウィンドウ値が大きくなり過ぎてしまうと、受信側の ACK 送信間隔と無関係に、送信側はデータセグメントをバースト的に送信してしまう可能性が高くなってしまふ。そこで、RSC-VAR では送信側の送信ウィンドウが大きくなり過ぎないように ACK セグメントの広告ウィンドウ値を利用する。受信側は ACK セグメントの広告ウィンドウ値を単純に受信バッファの空き容量とするのではなく、受信側で計算した送信レートに基づいた帯域遅延積と受信バッファの空き容量のうち小さい値を広告ウィンドウ値とする。広告ウィンドウ値 W は以下の式に従い算出される。

$$W = \min((srtt/T_{ack}) \cdot S_{mss}, S_{buf}) \quad (6)$$

S_{mss} は最大セグメント長, S_{buf} は受信バッファの空き容量である。

2.3 RSC-VAR の問題点

無線アドホックネットワークを有線ネットワークに接続した環境で RSC-VAR を利用した場合, 有線ネットワークの遅延が大きいと高いスループットを得ることができないという問題がある。

RSC-VAR は TCP-VAR の手法にならない, $vcwnd$, $srtt$ を用いて送信レートを決定する。これまでの RSC-VAR では, 式 (5) に示すように送信レート算出において $vcwnd$ を $srtt$ で割るため, 送信レートの値, ならびにその増減速度が sRTT に依存してしまう。sRTT の値が大きければ送信レートの値は小さくなり, かつその増減速度も小さくなる。

無線アドホックネットワークにおける適切な送信間隔は, 先に送信したセグメントが自己干渉を起こさない位置まで到達する時間であり, 送信レートの増減速度が無線アドホックネットワーク以外の部分も含む End-to-End の RTT に依存するべきではない。

有線ネットワーク側の遅延が長いと以下のような問題が生じる。有線側の遅延が大きい場合, $srtt$ が大きくなるので無線アドホックネットワークでの FHD から計算される送信レートに比べて送信レートが低くなることもある。また送信レートの増加速度が小さくなるので, 適切な送信レートになるまで長い時間が必要となってしまう。

3. 改良方式

3.1 基本戦略

前述の問題を解決するため, 送信レートの制御を RTT に依存しない改良版の RSC-VAR を提案する。改良版では, 新たに制御変数として基本送信間隔 (T_{base}) を導入し, 一定期間ごとに ACK 送信間隔を式 (7) に従って決定する。

$$T_{ack} = T_{base} \cdot \exp(VAR_{AT}) \quad (7)$$

T_{base} は過去 sRTT における VAR_{AT} の平均 ($\overline{VAR_{AT}}$) によって決定される。 $\overline{VAR_{AT}}$ が閾値より小さい場合に T_{base} は減少し, 閾値より大きい場合に T_{base} は増加する。また, T_{base} の制御はスロースタート状態と輻輳回避状態で異なる方法を用いる。 T_{base} の制御方法を式 (8), (9) に示す。

$$T_{base} = f(T'_{base}, \overline{VAR_{AT}}, s) \quad (8)$$

$$f(T'_{base}, \overline{VAR_{AT}}, s) = \begin{cases} \alpha \cdot T_{base} & (s = SS \wedge \overline{VAR_{AT}} < \theta_{ss,low}) \\ \beta \cdot T_{base} & (s = SS \wedge \overline{VAR_{AT}} > \theta_{ss,up}) \\ T_{base} - \gamma & (s = CA \wedge \overline{VAR_{AT}} < \theta_{ca,low}) \\ T_{base} + \delta & (s = CA \wedge \overline{VAR_{AT}} > \theta_{ca,up}) \end{cases} \quad (9)$$

ここで T'_{base} は前回設定された T_{base} の値である。 s はスロースタート状態 (SS) か輻輳回避状態 (CA) かを表す変数, $\theta_{ss,low}$, $\theta_{ss,up}$, $\theta_{ca,low}$, $\theta_{ca,up}$ は $\overline{VAR_{AT}}$ の閾値である。また, α , β , γ , δ は定数で $0 < \alpha < 1$, $\beta > 1$, $0 < \gamma$, $0 < \delta$ である。

T_{base} の増減判定に $\overline{VAR_{AT}}$ を用いるのは, ACK 送信間隔が長い場合に T_{base} が減少しにくくならないようにするためである。これまでの RSC-VAR では送信レートの増加判定基準を VAR_{AT} が N 回連続で閾値を下回ることにしていた。そのため, 送信レートが低い (つまり ACK 送信間隔が長い) 場合には, N 回データセグメントを受信するまでに時間がかかってしまい, 送信レートが増加しにくくなってしまう。一方, $\overline{VAR_{AT}}$ の値を用いることで, 送信レートが低い場合でも, 素早く T_{base} を減少させることが可能である。

3.2 T_{base} の初期化

T_{base} の初期値を自己干渉を起こさない程度に十分に大きな値とする。あるデータセグメントがネットワーク中に存在している時間は片方向遅延で, この値は RTT の 1/2 程度である。したがって, RTT の 1/2 は自己干渉を起こさない十分大きな ACK 送信間隔であるので, これを T_{base} の初期値とする。

TCP コネクション確立直後は RTT が測定できないので, 受信側では ACK 送信制御を行わず, データセグメント後直ちに ACK セグメントを送信する。受信側では通信開始後最初のデータセグメントに対する ACK に対する応答, つまり通信開始時から 2 つ目のデータセグメントを受信すると RTT を知ることができる。RTT (T_{rtt}) が測定できると, 式 (10) に従って T_{base} を初期化する。

$$T_{base} = 1/2 \cdot T_{rtt} \quad (10)$$

3.3 スロースタート

スロースタート状態では, T_{base} を指数関数的に減少させることで, ACK 送信間隔を適切な値に短時間で推移させる。

スロースタート状態では急激に T_{base} を変更するため, ネットワークが輻輳していない状態でもスループットの揺らぎが大きくなる傾向がある。そのため, 閾値 $\theta_{ss,low}$, $\theta_{ss,up}$ は大きめの値に設定する。

スロースタート状態において, 輻輳の兆候が検出さ

れるか ACK 送信間隔が適切な値になったと判断された場合、スロースタート状態から輻轉回避状態への移行する。改良手法では \overline{VAR}_{AT} が閾値 $\theta_{ss,up}$ を越えた場合、ネットワークが輻轉し始めたと判断し、 $T_{base} = \beta \cdot T'_{base}$ として輻轉回避状態へと移行する。また、一定期間新たなデータセグメントが受信できない (タイムアウト) 場合も同様に輻轉と判断して、 $T_{base} = \beta \cdot T'_{base}$ とし、輻轉回避状態へと移行する。本稿ではデータセグメントを受信してからタイムアウトまでの時間を $sRTT$ の4倍とした。

スロースタート状態で一定期間 T_{base} が変化しなかった場合、ACK 送信間隔が適切な値になったと判断し、輻轉回避状態へと移行する。この判断のための期間は n 回データセグメントを受信するまでとする。

3.4 輻轉回避

輻轉回避状態ではネットワークの状態に合わせて ACK 送信間隔を調節するために T_{base} を線形に推移させる。

輻轉回避状態では T_{base} の変動が少ないため、 T_{base} が適切な値であれば、スループットの揺らぎは小さい。したがって、閾値 ($\theta_{ca,low}$, $\theta_{ca,up}$) を小さな値とする。

ネットワークの状態が著しく変化した場合、 T_{base} を初期化して、輻轉回避状態からスロースタート状態への移行するべきである。無線アドホックネットワークでは無線での伝播エラーなど輻轉以外の要因でパケットロスが発生する。そのため、単純に通常の TCP と同様にパケットロスの発生をスロースタートへの移行条件とするべきではない。スロースタートへ移行する条件として、スループットの揺らぎの急激な増加や RTT の急激な変動などが考えられるが、これらについては検討が不十分であるため今後の検討課題とする。

3.5 T_{base} の更新間隔

T_{base} の増減判定は一定期間ごとに行う。受信側で T_{base} を更新してから、この変更が受信側でのデータセグメント受信に反映されるまでには時間差が存在する。そのため、データセグメント受信の度に T_{base} を更新すると ACK 送信間隔が不安定になってしまう可能性がある。例えば、スループットの揺らぎが小さくなると受信側で T_{base} を減少させる。このとき、ネットワークの遅延が大きいと、受信側で変更された ACK 送信間隔が実際のデータセグメントの受信に反映されるまでに長い時間がかかる。この間、受信側ではデータセグメントを受信する度に T_{base} を減少させるように処理が行われ続ける。この結果、ACK 送信間隔が短くなり過ぎ、スループットの揺らぎが大きくなってしまい、受信側では T_{base} を増加させることになる。そし

表 1 実験環境

Simulator	QualNet 4.0
Wireless MAC	IEEE 802.11b
Wireless BandWidth	2 Mbps
Communication Range	340 m
Carrier sense Range	540 m
Wired MAC	IEEE 802.3
Wired Delay	200 ms
Wired BandWidth	10 Mbps
Sender TCP	NewReno
MSS	1460[byte]

表 2 RSC-VAR (改良手法) のパラメータ

$\theta_{ss,up}$	0.3	n	50
$\theta_{ss,low}$	0.2	α	0.9
$\theta_{ca,up}$	0.15	β	1.2
$\theta_{ca,low}$	0.05	γ	1 ms
K_{normal}	50	δ	3 ms
$K_{congestion}$	5		
TimeOut	$4 \cdot sRTT$		

表 3 RSC-VAR のパラメータ

θ_{upper}	0.3	N	300
θ_{lower}	0.2	N_{slow}	50
K_{normal}	50	M	50
$K_{congestion}$	5	α'	1.0
		β'	0.5

て、また T_{base} 減少時と同様の理由により ACK 送信間隔が減少しすぎてしまう。このようにして送信レートは変動し続ける。

受信側での T_{base} の変更がネットワークに反映され、受信側で観測できるようになるまでは少なくとも RTT を要する。さらに、 \overline{VAR}_{AT} の計算には過去 $srtt$ の受信データを用いるため、新たな T_{base} に対する \overline{VAR}_{AT} を計算するためには $2 \cdot srtt$ 程度必要である。よって、 T_{base} の更新は $2 \cdot srtt$ ごとに行う。

4. シミュレーション

4.1 シミュレーション条件

RSC-VAR の有効性を検証するためにシミュレーション実験を行った。シミュレータに QualNet4.0 を使い、FTP による通信を行ったときのスループットを評価した。実験環境、RSC-VAR (改良手法)、RSC-VAR の各パラメータは表 1、表 2、表 3 に示す通りである。

RSC-VAR の改良手法の評価のため、図 4 に示すように一直線上に 5 台の端末を 300m 間隔で配置し、一

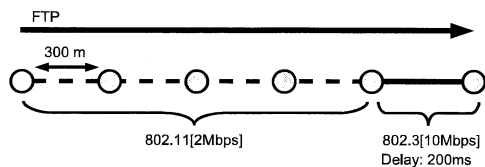


図4 シミュレーションに用いたネットワーク

端の端末をゲートウェイとし有線ネットワークに接続したネットワークを想定した。無線アドホックネットワーク上の端末と有線ネットワーク上の端末間で、シミュレーション開始後1sからシミュレーション終了までFTP通信を行う。シミュレーション時間を300sとし、送信側TCPをTCP-NewRenoとした。また、受信側TCPにはRSC-VARの送信レート制御を改良したRSC-VAR(改良手法)、比較対象としてオリジナルのRSC-VAR、NewRenoの3種を用いた。各モデル毎にシミュレーションを5回行い、スループットの時間推移を評価した。

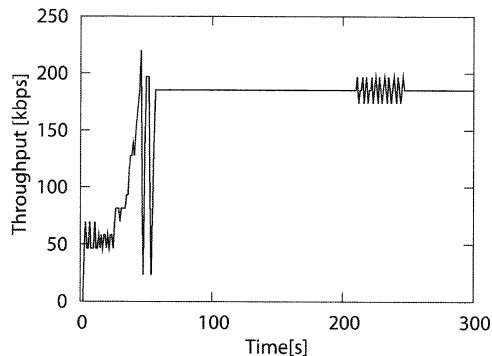
4.2 シミュレーション結果

図5(a), (b), (c)にスループットの時間推移を示す。グラフは5回の試行のうちの一例である。また、他4回の試行においてもほぼ同様の傾向が見られた。

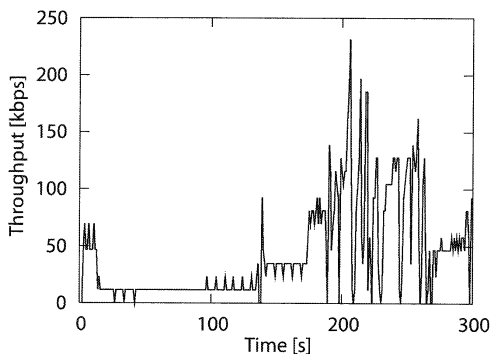
図5(a)より、RSC-VAR(改良手法)はシミュレーション開始後60sからシミュレーション終了まではスループットが185kbps程度で安定していることが分かる。シミュレーション時間中の平均スループットは約167kbpsであった。一方、図5(b)より、改良前のRSC-VARはシミュレーション開始後20sから140sまでスループットが低い値を推移しており、180sから270sまではスループットが激しく変動している。シミュレーション時間中のスループットの平均は約42kbpsで、RSC-VAR(改良手法)の25%程度である。図5(c)に示すNewRenoの場合には、瞬間的に250kbpsと高いスループットが得られているが、全体としてスループットは激しく振動している。シミュレーション時間中の平均スループットは約119kbpsであり、RSC-VAR(改良手法)の70%程度である。

図6にRSC-VAR(改良手法)における実際のACK送信間隔の推移を示す。図6中の破線は予備実験によって求められた最適なACK送信間隔である。予備実験では T_{ack} の値を固定にした場合のスループットを、5~50msまでの様々な T_{ack} に対して求めた。

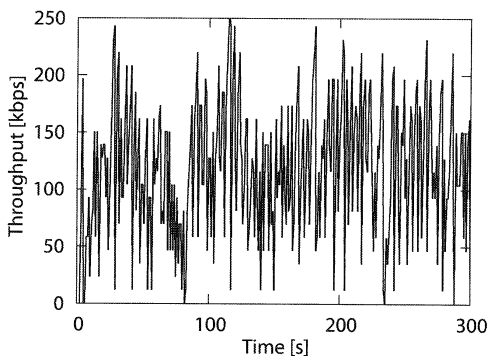
図6を見ると、RSC-VAR(改良手法)におけるACK送信間隔 T_{ack} は、シミュレーション開始後60sから



(a) RSC-VAR (改良手法)



(b) RSC-VAR



(c) NewReno

図5 スループットの推移

シミュレーション終了までの期間では約55msを維持している。このときの送信レートは約212kbpsである。前述の通り、この間に得られたスループットは約185kbpsである。このスループットの値と予備実験で得られたスループットの最大値(約200kbps)との差は10%以下である。また、このときの T_{ack} の値は予備実験で得られた最適値よりも22%程度大きい。

図7にRSC-VAR(改良手法)における T_{base} の推移

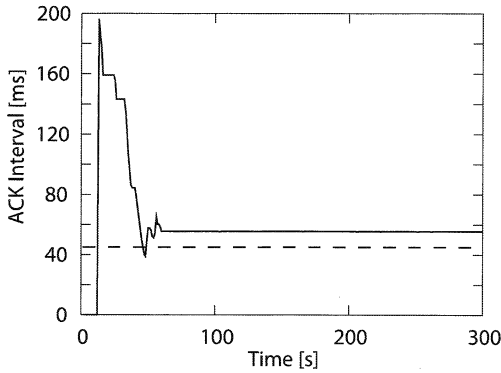


図 6 RSC-VAR (改良手法) における ACK 送信間隔の推移

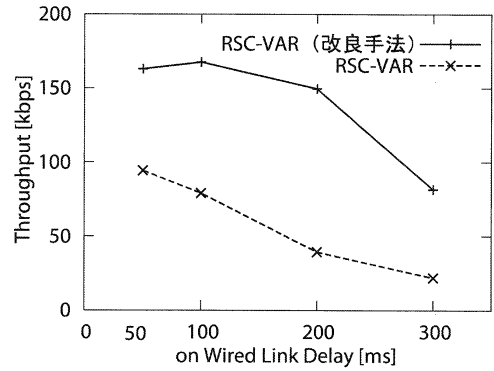


図 9 RSC-VAR (改良手法) における有線遅延と平均スループット

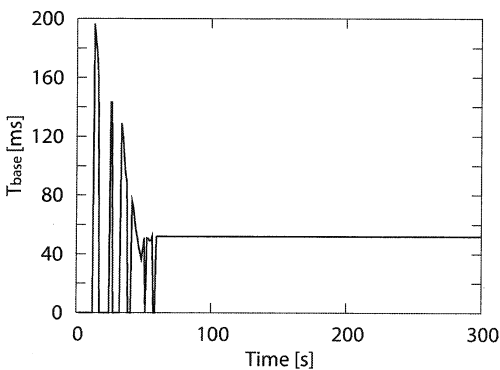


図 7 RSC-VAR (改良手法) における T_{base} の推移

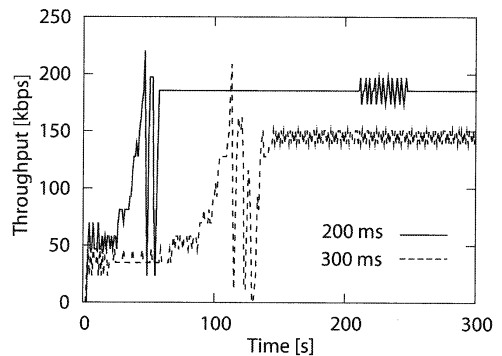


図 10 有線遅延 300 ms における RSC-VAR (改良手法) のスループットの推移

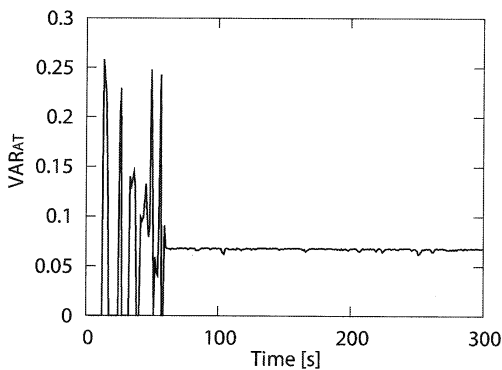


図 8 RSC-VAR (改良手法) におけるスループットの揺らぎの推移

を示す。シミュレーション開始後約 50 s でスロースタート状態から輻輳回避状態へ移行していることが分かる。この状態変化は、パケットロスにより受信側でのデータセグメントの受信がタイムアウトしたために起こったものである。なお、RSC-VAR (改良手法) ではパケットロスはシミュレーション開始後 45 s から

55 s までの間に 3 つ発生したのみである。これに対し RSC-VAR は 92 回、NewReno は 246 回のパケットロスが発生している。

RSC-VAR (改良手法), RSC-VAR について有線ネットワークの遅延を 50 ms, 100 ms, 200 ms, 300 ms と変化させ、それぞれにおいて同様にシミュレーションを行った。各条件における平均スループットを図 9 に示す。同図より RSC-VAR (改良手法) は全領域にわたって RSC-VAR に比べ高いスループットが得られているが、有線側の遅延を大きくしていくと、スループットが減少する傾向にあることが分かる。これは RTT が長くなったことで、スロースタート状態における ACK 送信間隔の減少速度が低くなってしまったためと考えられる。図 10 に有線側の遅延を 200 ms および 300 ms とした場合のスループットの推移の一例を示す。両者を比べると、全体の形状はほぼ同様の傾向が見られるが、スループットが高い値になるまでの時間が有線遅延 300 ms の場合では明らかに長くなってしまっ

いる。

4.3 考 察

RSC-VAR の改良手法は改良前に比べ有線側の遅延が長いネットワークで高いスループットを得られた。しかし、まだ有線遅延の増加に伴い得られるスループットが減少してしまう傾向が強くみられる。これは、RTT の増加に伴い T_{base} の更新間隔が長くなってしまったため、スロースタート状態のときの T_{base} 減少速度が低くなってしまったためと考えられる。そのため、今後、更なる RSC-VAR の性能向上のため、スロースタート状態における T_{base} の制御方法について検討が必要である。

また、 AT や $\overline{VAR_{AT}}$ の計算の際にも sRTT を利用しているため、これらの影響についても今後、検討が必要である。

5. ま と め

本稿では、無線アドホックネットワークにおける TCP のための受信側端末主導のデータ送信レート制御手法 RSC-VAR (Receiver initiated Sending-rate Control based on TCP-VAR) の問題点として、遅延が大きなネットワークにおいて高いスループットが得られないことを示し、その改良手法を提案した。RSC-VAR は送信レート決定に RTT を利用しているため、無線アドホックネットワークをインターネットにつなげた場合、インターネット側の遅延の影響を受けてしまい送信レート制御が有効に働かなかった。この問題を解決するため、改良手法では送信レートの決定に RTT の値そのものを用いずに、基本送信間隔 T_{base} を用いて ACK 送信間隔を制御する。

改良手法をシミュレーションにより評価した結果、無線アドホックネットワーク内にゲートウェイを設置し、有線ネットワークにつなげた直線型トポロジにおいて単一の FTP フローを流した場合にスループットが向上することを確認した。有線側の遅延を 200 ms とした場合、改良手法は改良以前に比べ約 4 倍のスループットを得ることができた。しかし、改良手法は、有線側の遅延をさらに長くした場合に高いスループットを得るまでに長い時間を要するという問題点があることが分かった。この原因として、改良手法では、 T_{base} の更新間隔や $\overline{VAR_{AT}}$ の計算において RTT の影響を間接的に受けやすくなることが考えられる。

今後の課題として、RTT に依存しない ACK 送信間隔の増減判定方法の改良や他フロー、中継経路の変更へ対応することがあげられる。また、さまざまなトポロジでの RSC-VAR の評価、送信側が TCP-VAR のよ

うな送信レート制御を行う場合の RSC-VAR の挙動についても確認する予定である。

参 考 文 献

- 1) Holland, G. and Vaidya, N.: Analysis of TCP Performance over Mobile Ad Hoc Networks, *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking, MobiCom 1999*, pp.219–230 (1999).
- 2) Fu, Z., Zeros, P., Luo, H., Luo, S., Zhang, L. and Gerla, M.: The Impact of Multihop Wireless Channel on TCP Throughput and Loss, *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2003*, Vol.3, pp.1744–1753 (2003).
- 3) ElRakabawy, S.M., Klemm, A. and Lindemann, C.: TCP with Adaptive Pacing for Multihop Wireless Networks, *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing, MobiHoc 2006*, pp.288–299 (2005).
- 4) Chen, J., Gerla, M., Lee, Y.Z. and Sanadidi, M.: TCP with Variance Control for Multihop IEEE 802.11 Wireless Networks, *Proceedings of Military Communications Conference 2006, MILCOM 2006*, pp.1–7 (2006).
- 5) 小嶋明寿, 石原 進: アドホックネットワークにおけるデータ受信レートに基づく受信側 TCP 主導によるデータ送信レート制御に関する一検討, 電子情報通信学会技術研究報告, 情報ネットワーク, Vol.107, No.483, pp.59–64 (2008).
- 6) 小嶋明寿, 石原 進: アドホックネットワークにおけるデータ受信レートに基づく受信側 TCP 主導によるデータ送信レート制御, マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO2008) シンポジウムアブストラクト集, Vol.2008, No.1, pp.892–899 (2008).