3U-3

モバイルアドホックネットワークのための中継端末の送信バッファ残量を用いた輻輳ウィンドウの初期化方法の提案

小嶋明寿 † 石原進 ‡

† 静岡大学工学部 ‡ 静岡大学創造科学技術大学院

1 はじめに

MANET では端末の移動により通信経路の変更，分断によるパケットロスが頻発する．このため MANET 上で TCP 通信を行うと経路変更に伴うパケットロスが発生し，タイムアウトが発生する．タイムアウトが発生すると送信側の TCP は輻輳状態と判断し，cwnd を減少させる．従って cwnd が大きくならず，スループットが低下してしまう．この問題に対し Holland らは，cwnd の減少を防ぐためリンク切断を TCP に明示的に報せる ELFN[1] を用いた輻輳制御手法を提案した．また，Yu は ELFN を拡張しパケットロスを通知する EPLN[2] を用いる手法を提案した．これらの手法は経路切断を検知した後，輻輳制御を無効にすることで cwnd の減少を防ぐ．しかし通信再開後の cwnd に対する考慮はされていない．そのため通信再開直後に新たな経路上で輻輳を誘発する恐れがある．

この問題を解決するため，本稿では通信開始時及び経路変更後の cwnd を決定する手法として中継端末の送信バッファ残量を考慮した cwnd 初期化手法 (RTBCC: Rest of Transmission Buffer based Congestion window Control) を提案する．これにより通信開始直後から高いスループットを得ること，経路長が異なった複数のフローが交差した場合の公平性の向上を目指す．

2 RTBCC

経路変更に伴うタイムアウトが発生した場合，必ずしも cwnd を小さくする必要はない．しかし，輻輳が発生した場合は輻輳の程度にあわせて cwnd を小さくする必要がある．スループットを向上させるためには，利用経路を輻輳させない程度に大きな cwnd 初期値が必要である．そこで RTBCC では cwnd の初期値の指標として中継端末の送信バッファ残量 (Rest of Transmission Buffer(RTB)) の最小値を用いる．cwnd を RTB を用いて初期化することで通常より大きな値に設定することが可能となる．また RTB をネットワークの許容量を示す値として利用することで経路を輻輳させる危険性を低くすることが期待できる．

2.1 cwnd 初期値の決定

RTBCC では cwnd を初期化するために送信端末は経路上の端末の RTB の最小値を知る必要がある．中継端末の RTB は端末がパケットを中継するときに転送パケットに付加される．RTB の獲得方法については次節に述べる．以下に中継端末より報告された RTB を用いた cwnd 初期値決定手法を述べる．

RTB の最小値をネットワーク許容量と比較した場合，この値はネットワーク許容量以下であることが期待できる．従って RTB の最小値で cwnd を初期化することで輻輳を起こさない程度の通信を行うことができると考える．

通信経路上に存在する通信フローは 1 つとは限らない．複数の通信フローが存在した場合，中継端末の送信バッファを他のフローと共有することになる．そこで cwnd の初期値 ($cwnd_{\mathrm{init}}$) を以下の式に従い求める．

$$cwnd_{\mathrm{init}} = \alpha \cdot \frac{RTB_{\mathrm{min}}}{flow}$$

RTB_{min} は報告された RTB の最小値，$flow$ は交差しているフロー数，α はネットワーク利用率を決定する係数で $0 < \alpha \leq 1$ の範囲をとる．α を大きくすると cwnd の値は大きくなるが輻輳を誘発する危険も大きくなる．また，小さくすれば輻輳は起きにくくなるが cwnd が小さくなってしまいスループット向上が期待できない．

2.2 RTB，通信フロー数の獲得

送信端末が経路上の端末の RTB の最小値，経路上の通信フロー数の最大値を獲得するために，各中継端末は転送パケットに自身の RTB，通信フロー数を付加するための特別な機能を追加する．また，パケットに各情報を付加するためにパケットヘッダに専用の領域を設ける．

RTB 報告のためにパケットの IP ヘッダに専用の領域 (報告フラグ，RTB 報告フィールド，フロー報告フィールド) を設け，この領域は TCP から可視であるとする．各中継端末は報告フラグが付いたパケットを中継するとき，自身の RTB，この端末を利用している通信フローの数を確認する．転送パケットの RTB 報告フィールド値より自身の RTB が小さい場合，このフィールドに自身の RTB を書き込む．同様に転送パケットの

Proposal of Initialization Congestion Window Based on the Rest of Transmission Buffer on Intermediate Node for MANET

Akihisa KOJIMA†, Susumu ISHIHARA‡

†Faculty of Engineering, Shizuoka University ‡Graduate School of Science and Technology, Shizuoka University

表 1: シミュレーション環境

シミュレータ	ns-2
通信インターフェース	IEEE802.11
通信半径	250 m
通信レート	1 Mbps
インターフェースキュー	50 packet
シミュレーション時間	100 s
アプリケーション	FTP

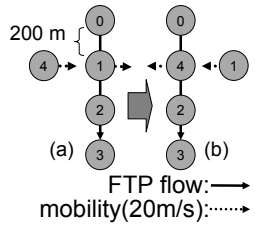

図 1: シナリオ 1

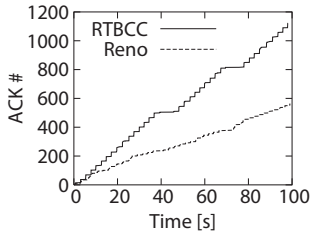

図 2: トポロジ変化時の受信 ACK 番号の変化

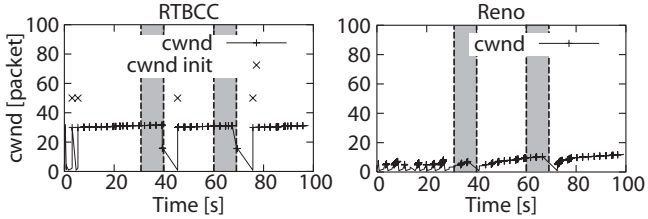

図 3: トポロジ変化時における cwnd の変化

フロー報告フィールド値と自身の値を比較し，自身の値が大きい場合にこのフィールドを更新する．

送信端末はタイムアウトが発生するとその後の最初のデータパケット送信時に報告フラグを有効にする．受信端末は報告フラグが有効になっているパケットを受信すると，そのパケットに対する ACK の IP ヘッダ内の報告フラグを有効にし，各報告フィールドに受信したパケットの報告フィールドの値をコピーする．この ACK パケットを受信した送信端末は各フィールド値を基に cwnd を初期化する．

また，コネクション開始時には SYN セグメントを格納するパケットの報告フラグを有効にすることで，上記と同様の手順で RTB を獲得する．

3 実験と考察

3.1 シミュレーション条件

RTBCC の効果を評価するためにシミュレーション実験を行った．シミュレーション環境を表 1 に示す．提案手法である RTBCC は cwnd 初期化部分以外は TCP/Reno と全く同様の動作を行う．また，RTBCC のパラメータ α を 0.6 とした．

3.2 トポロジが変化する場合

端末の移動によりトポロジが変化するシナリオで実験を行った．図 1 の (a) の状態から端末 1，4 が移動して (b) の状態になる．その後，端末 1，4 がさらに移動して (a) の状態に戻る．(a) → (b) の移動は 30 秒から 40 秒にかけて行い，(b) → (a) は 60 秒から 70 秒にかけて行う．図 2，図 3 にこのときの送信側における受信 ACK 番号の変化および cwnd の変化を示す．図 3 のグレー部分は端末の移動を示す．図 2 の 40，70 秒付近で ACK 番号の増加率がほぼ 0 となっているのはトポロジの変化に伴ったパケットロスが発生したためである．しかし，RTBCC は ACK 番号の増加率がパケットロス後すぐに元に戻っている．これは RTBCC がスロースタート域を必要としないためである．また，図 3 では Reno は cwnd が小さな値のままであるのに対し，RTBCC は cwnd が大きな値で安定している．このため，図 2 に見られるように RTBCC の ACK 番号増加率は Reno よりも高い．これらの結果から RTBCC は経路変更が発生し易い場合において有効であるといえる．

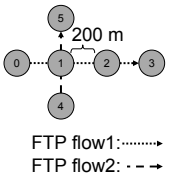

図 4: シナリオ 2

表 2: 十字型トポロジにおける各フローのスループット

Throuput (kbps)	Flow 1	Flow 2	Total
RTBCC	34.00	3.06	37.06
Reno	39.10	0.92	40.02

3.3 経路長が異なった複数のフローが交差する場合

複数のフローが存在する場合の性能を調べるため図 4 のような十字型のトポロジ上で複数のフローが交差する場合について実験をした．表 2 に 10 回の実験による各フローの平均スループットを示す．表 2 より，Reno では Flow 2 は Flow 1 の約 40 倍のスループットで通信しており，フロー間に公平性がない．これに対し，RTBCC は Flow 2 は Flow 1 の約 11 倍のスループットが得られており，Reno よりもフロー間の公平性が向上している．各フローのスループットの合計は Reno が約 40kbps，RTBCC が約 37kbps であり，全体の性能に差はほとんど見られない．

4 まとめ

MANET 上での TCP 通信における送信レート制御手法として cwnd の初期値に RTB の最小値を利用する RTBCC を提案した．シミュレーションの結果，トポロジが頻繁に変化する環境での有効性が確かめられた．また，経路長の異なる複数のフローが中継端末を共有する場合のフロー間の公平性の向上を確認した．今後，様々なトポロジにおける性能評価，複数のフローが共存していた場合の公平性についての詳細な検討，ELFN，EPLN との併用によるスループットの向上についての検討を進める予定である．

参考文献

[1] Gavin Holland and Nitin Vaidya : “Analysis of TCP Performance over Mobile Ad Hoc Networks”, MobiCom’99, pp.219-230, 1999

[2] Xin Yu : “Improving TCP Performance over Mobile Ad hoc Networks by Exploiting Cross-Layer Information Awareness”, MobiCom’04, pp.231-244, 2004