

# 無線メッシュネットワークにおける経路変更に起因する冗長な TCP 高速再送制御の抑制手法

明田 修平<sup>†a)</sup>      瀧本 栄二<sup>†</sup>      大月 勇人<sup>†\*</sup>      齋藤 彰一<sup>††</sup>  
毛利 公一<sup>†</sup>

A Method to Suppress Redundant TCP Fast Retransmit Caused by Route Change in Wireless Mesh Networks

Shuheï AKETA<sup>†a)</sup>, Eiji TAKIMOTO<sup>†</sup>, Yuto OTSUKI<sup>†\*</sup>, Shoichi SAITO<sup>††</sup>, and Koichi MOURI<sup>†</sup>

あらまし 無線メッシュネットワークはリンク品質が安定しにくいいため、経路変更によってリンク品質の変化に対応することで通信性能を向上させることができる。しかし、経路変更は順序エラーの原因となり、特に TCP 通信における冗長な高速再送制御を引き起こすという課題がある。この課題を解決するため、本論文では、輻輳によるキュー溢れを検知するための TCP ヘッダの改良及び受信 DATA パケットのシーケンス番号と到着間隔とに着目した順序エラー発生原因の判定手法と、経路変更により発生した順序エラーによる冗長な高速再送制御の抑制手法を提案する。提案手法では、判定手法により順序エラーの原因が経路変更であると判定すると、ACK パケットの送信を一時待機することで冗長な高速再送制御を抑制する。また、誤判定に対しても待機時間に制限を設けることで影響を最小限に抑えている。提案手法をシミュレーション評価した結果、冗長な再送パケット数を 70%~99%削減でき、スループットも 1~5%向上することを確認した。

キーワード 無線メッシュネットワーク, TCP, 順序エラー, 高速再送制御, 輻輳制御

## 1. ま え が き

通信インフラストラクチャの構築が困難な環境や、災害時などに通信環境を提供する手段として、無線メッシュネットワークが注目されている。無線メッシュネットワークは、各ノードが直接無線で接続してルータとして動作することでネットワークを構築する。しかし、無線メッシュネットワークは、衝突や干渉などによる影響を受けやすいという欠点がある。また、各ノードが自律分散的に通信を行う無線メッシュネットワークには、トラヒックが集中する経路とそうでない

経路が混在しやすい特徴がある。そのため、通信に最適な経路は、各ノードの通信状況に応じて変動しやすい。この特徴から、無線メッシュネットワークでは、リンクステート型ルーティングプロトコルを用いた通信経路制御が最適とされている [1], [2]。リンクステート型ルーティングプロトコルは、周期的に制御メッセージを送受信することでトポロジーのリンク関係やリンク品質などの情報を収集し、それらの情報をもとに最適な経路を構築する。しかし、リンクステート型ルーティングプロトコルを用いた場合、経路変更の頻度が増加し、先行パケットの追越しによる到着順序エラー（以下、順序エラーとする）が発生することが報告されている [3]~[5]。

経路変更による順序エラーが発生する典型例を図 1 を用いて説明する。図 1(a) では、時計回りの経路が通信開始時に選択されているものとする。時間経過とともに、周囲からの電波干渉や隠れ端末問題による影響により当該経路のリンク品質が低下すると、反時計回りの経路へ経路変更が行われる（図 1(b)）。このとき、

<sup>†</sup> 立命館大学情報理工学部, 草津市  
College of Information Science and Engineering,  
Ritsumeikan University, 1-1-1 Noji-higashi, Kusatsu-shi,  
525-8577 Japan

<sup>††</sup> 名古屋工業大学, 名古屋  
Nagoya Institute of Technology, Gokiso-cho, Showa-ku,  
Nagoya-shi, 466-8555 Japan

\* 現在, NTT セキュアプラットフォーム研究所

a) E-mail: saketa@asl.cs.ritsumei.ac.jp

DOI:10.14923/transcomj.2015JBP3066

新たな経路の中継ノードのキューには待機中のパケットがないことやホップ数の差から、受信ノードまでの到着時間が短くなる。その結果、変更前の経路と変更後の経路との到着時間の差から、4番目のパケットよりも9, 10, 11番目のパケットが先に到着するパケットの追越しが発生する。トランスポートレイヤプロトコルのTCPには順序エラー訂正機能がある。しかし、TCPは、順序エラーの訂正に必要なパケットの到着が遅れると、パケットの消失が発生したと判断して高速再送制御を行う。経路変更に伴う順序エラーは、混雑した経路から空いている経路への変更によって発生するため、パケットの追越し幅（追越ししたパケットの数）が大きくなる傾向がある[6]。そのため、順序エラー発生時の追越し幅が高速再送制御の起動条件を満たすと、再送の対象となるパケット（図1の4番目以降の一部のパケット）が再送される。しかし、経路変更に伴う順序エラーの場合、パケットは消失していないため再送しなくても訂正が可能であるため、この高速再送制御は冗長であるといえる。更に、このとき輻輳制御も同時に行われており、ウィンドウサイズを低下させ送信速度が低下する。しかし、実際には空いた経路を用いるため、輻輳制御を行う必要はない。

以上から、本論文では、無線メッシュネットワーク上でのTCP通信を対象に、経路変更によって発生する冗長な高速再送制御を抑制する手法を提案する。冗長な高速再送制御はTCPがもつ次の2点の特徴によって発生する。

- 受信ノードは順序エラー発生時に、その原因にかかわらず重複ACKとなるパケットを送信ノードに返送する。
  - 送信ノードは同一シーケンス番号をもつ重複ACKを三つ受信すると無条件で高速再送制御を行う。
- 上記の2点は、重複ACKを通じて関連している。この点に着目すると、経路変更に起因する順序エラー

に対して、受信ノードが重複ACKを返信しないことで、冗長な高速再送制御を抑制できると考えられる。そこで、提案手法では、まず、受信ノードにおいて順序エラーを検知した際にその発生原因を判定する。具体的には、無線環境特有の電波干渉などによるMACレイヤ以下でのパケットの消失（以下、パケットロスとする）、輻輳時のキュー溢れによるネットワークレイヤでのパケットの消失（以下、パケットドロップとする）、及び経路変更の3種類のうちどれであるかを判定する。発生原因が経路変更であると判定した場合は、受信ノードにおいて重複ACKとなるACKパケットの送信を待機することで高速再送制御を抑制する。経路変更以外の原因と判定した場合は、通常のTCPに則った処理を行い、送信ノードによる高速再送制御の実行を促す。これらにより、提案手法は、冗長な高速再送制御を抑制することで、不必要な再送制御と輻輳制御を抑制でき、その結果として効率的な通信帯域の利用が期待できる。

## 2. 予備実験

無線メッシュネットワークにおける順序エラー発生時の挙動、及び発生原因ごとの傾向を明らかにするために、ネットワークシミュレータであるQualNet [7]を用いて予備実験を行った。予備実験では順序エラーの発生原因ごとのシーケンス番号と到着間隔の傾向を調査した。予備実験で得られた傾向から、これらにしきい値を設定し順序エラーの原因判定方法について考察する。

### 2.1 シミュレーション環境

シミュレーションパラメータを表1に示す。シミュレーションを行うに当たり、QualNetに動的メトリックの一つであるETX [8]を実装し、かつOLSRv2 [9]がETXを利用できるように拡張した。これは、無線メッシュネットワークにおいて動的メトリックを採用した

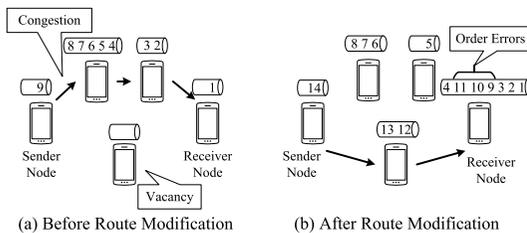


図1 順序エラー発生の典型例  
Fig. 1 Typical example of order errors.

表1 シミュレーションパラメータ  
Table 1 Simulation parameters.

パラメータ	設定値
伝搬モデル	2-ray モデル
シミュレーション時間	200 秒
インタフェース	IEEE802.11a
伝送速度	54Mbps
最大通信距離	50m
RTS/CTS	無効
パケットサイズ (MTU)	1500 バイト
MSS	1460 バイト
TCP	NewReno, SACK
ルーティングプロトコル	OLSRv2-ETX

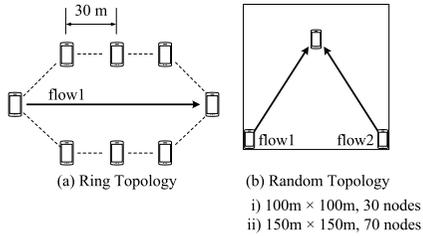


図2 評価用トポロジー  
Fig.2 Topologies for evaluation.

リンクステート型ルーティングプロトコルが適しているためである [1], [2]. OLSRv2 のパラメータは, RFC 7181 [9] で示された値とした. TCP には, Linux など一般的に用いられている輻輳制御アルゴリズムの中で QualNet で利用できる最新の TCP-NewReno [10] (以下, NewReno とする) と, SACK オプション [11] を有効にした TCP-SACK (以下, SACK とする) の二つを用いた. TCP の送受信バッファ及びインタフェースキューのサイズは 512KB とした. これは, RTT が 50 ms から 100 ms 程度であったことから, 帯域遅延積に基づき伝送速度 (54Mbps)  $\times$  RTT(70ms) から設定した. この設定に当り, TCP のウィンドウスケールリングオプション [12] を有効にした.

トポロジーは, 図 2(a) のリングトポロジーと (b) のランダムトポロジーを用いた. リングトポロジーは, 経路変更起因する順序エラーが発生しやすいことから, その特徴を観測するために用いた. ランダムトポロジーは, 本論文の対象である無線メッシュネットワークを想定したものであり, 複数のフローが 1 台のゲートウェイとして動作するノードに集中する環境を想定している. ランダムトポロジーは, 図 2 のように 3 台の送受信ノードの配置を固定とし, それ以外のノードの配置をランダムにした. また, ランダムトポロジーでは, ネットワークサイズによる影響を排除するため, 100 m  $\times$  100 m に 30 台のノード (以下, ランダム 100 トポロジーとする), 150 m  $\times$  150 m に 70 台のノード (以下, ランダム 150 トポロジーとする) を配置したサイズが異なる二つのトポロジーを用いた. アプリケーションは FTP とし, 一つのフローで 14.48MB (10,000 パケット相当) のファイルを送信した.

## 2.2 順序エラーに対する典型的な TCP の挙動

経路変更による順序エラー発生時の挙動を図 3 の TCP シーケンス番号の遷移例を用いて説明する. 図

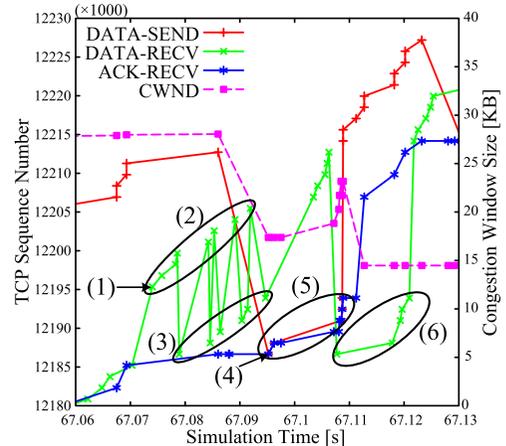


図3 順序エラー発生時のシーケンス番号の推移例  
Fig.3 Transition of sequence number with order errors.

3 はリングトポロジーの環境で得られた結果のうち通信開始後 67.06 秒~67.13 秒を抜粋したものである. DATA-SEND は送信ノードが送信した DATA パケットのシーケンス番号の推移, DATA-RECV は受信ノードで受信した DATA パケットのシーケンス番号の推移, ACK-RECV は送信ノードで受信した ACK パケットのシーケンス番号の推移をそれぞれ表す. CWND は, 送信ノードが NewReno により制御する輻輳ウィンドウサイズの推移を表す.

図 3 では (1) の箇所では選択経路のリンク品質が低下する. 具体的には, ETX のリンク品質計測用パケット (OLSRv2 の HELLO メッセージ) が隠れ端末問題による影響によって消失し, ETX によるリンク品質計測結果が悪化する. その結果, 経路変更が発生し, 6 パケットの追越しが発生し大きな差分のシーケンス番号をもつ DATA パケットを受信している. 以降, (2) の新経路を経由した八つの DATA パケットと, (3) の旧経路を経由した六つの DATA パケットが交互に受信ノードに到着している. 新経路を経由して到着した DATA パケットは順序エラーとなるため, これらに対する ACK パケットは重複 ACK として扱われる. この例では, 旧経路を経由した DATA パケットが遅れて到着するまでに, 八つの重複 ACK が送信されている. (4) で重複 ACK を 3 回連続して受信した送信元ノードは, 高速再送制御を行っている. このとき受信した ACK パケットには, SACK オプション [11] が付与されている. 送信側 TCP は, SACK で示された情

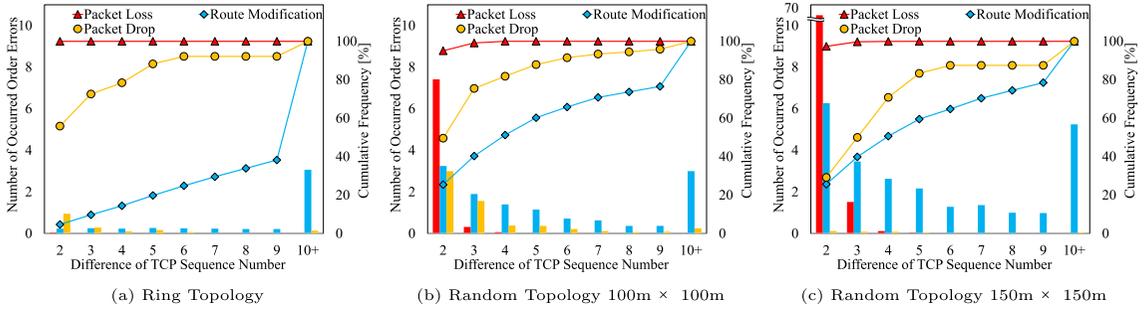


図 4 シーケンス番号の差分ごとの順序エラー発生数  
Fig. 4 Number of order errors for each difference of sequence number.

報に基づき、(5) で五つの DATA パケットを再送している。これらの再送パケットは、(6) で到着している。しかし、これらのパケットと同一シーケンス番号をもつパケットは消失せず (3) で受信していることから、この再送制御は冗長である。また、高速再送制御を行ったとき、輻輳ウィンドウサイズは、輻輳制御により  $ssthresh$  の値まで低下している。更に、順序エラーから回復していないときに、重複 ACK を受信すると輻輳制御により  $ssthresh$  の値まで低下させる。これらの輻輳制御は不必要である。

このように、効率的な再送を実現するために SACK オプションを使用した場合でも、SACK は、経路変更による順序エラーをパケットの消失によるものと誤認し、冗長な高速再送制御を行うことが確認できた。また、NewReno においても同様の冗長な高速再送制御が確認された。特に NewReno は、1 回のウィンドウサイズ分の送信に対して複数のパケットが消失する場合を想定し、再送時に複数の DATA パケットを再送するため、冗長な再送パケット数が SACK に比べて多いことが明らかとなった。

### 2.3 発生原因ごとのシーケンス番号の特徴

経路変更による順序エラーは、先行する DATA パケットを空いている新経路を通過する後続の DATA パケットが追い越して到着することで発生する。そのため、シーケンス番号の差分に着目すると、差分は追い越されたパケット分となり、大きくなることが予想される。このことは図 3 からも確認できる。一方で、パケットの消失は散発的若しくはバースト的に発生する。また、その原因は干渉などによるパケットロスと輻輳によるキュー溢れで発生するパケットドロップとに分類することができる。そのため、経路変更による順序エラーとは異なる特性を示すと考えられる。

そこで、シーケンス番号の差分ごとの順序エラーの発生数の統計調査を行った。その結果を図 4 に示す。横軸は、シーケンス番号の差分を MSS (Maximum Segment Size) で除算した値、すなわちシーケンス番号の差分を 1 パケット単位にした値を表す。シーケンス番号の差分は、DATA パケット受信時において TCP ヘッダのシーケンス番号と、直前に受信した DATA パケットのそれとの差分から求める。MSS で除算するのは、Nagle アルゴリズム [13] によってセグメントサイズが MSS となるように調整されるためである。なお、差分 1 は正常な順序となるため、除外した。10 パケット以上の差分は、10+ と表記する。主縦軸は NewReno 及び SACK をそれぞれ 30 回行った平均順序エラー発生数 (棒グラフ)、第二縦軸はその累積頻度 (折れ線グラフ) とした。各グラフは順序エラー発生原因ごとに分類している。

パケットロスが発生した際の差分は最大で 6 であり、各トポロジーで 95% 以上が差分は 2 であった。このことから、ほとんどのパケットロスは散発的に発生したことが分かる。一方、経路変更の場合は、差分の大きさに反比例して順序エラーの発生数が単調減少する傾向が見られた。その中でも差分が 3 以上は、リングトポロジーで 95%、ランダム 100 トポロジーで 75%、ランダム 150 トポロジーで 75% を占めていた。また、パケットドロップの場合の差分は、パケットロスに近い傾向があるものの大きな値となることもあり、中庸な性質を示した。ただし、発生数が他と比較して少なかった。

### 2.4 発生原因ごとの到着間隔の特徴

パケットロスによる順序エラーが発生すると本来到着すべき DATA パケットが到着しないため、DATA パケットの到着間隔が変動すると予想される [6]。一方

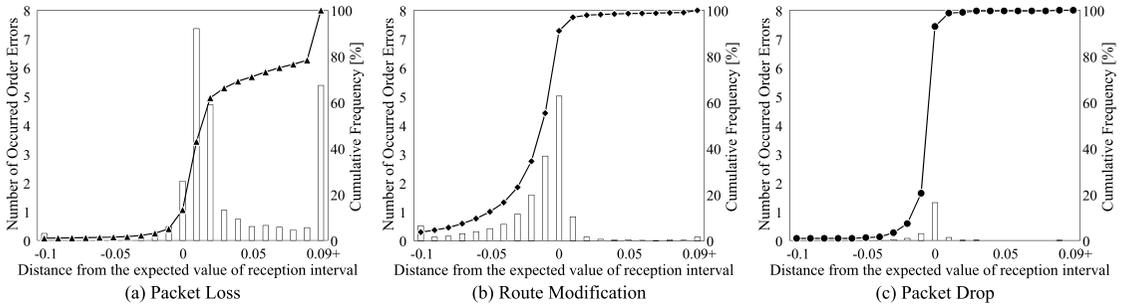


図5 到着間隔の期待値からの距離ごとの順序エラー発生数  
Fig. 5 Number of order errors for distance from the expected reception interval.

で、経路変更による順序エラーではパケットの消失が発生しないため、DATAの到着間隔に変化がないと予想される。

そこで、順序エラー発生時のDATAパケットの到着間隔の統計調査を行った。到着間隔はトポロジーや周辺の通信状況により変化するため、以下の式(1)で得られる*i*番目に観測されたDATAパケットの、到着間隔の期待値からの距離をヒストグラムとした。

$$D_i = T_i - T_{ave.} \times Diff_i \div MSS \quad (1)$$

$D_i$ は期待値からの距離、 $T_i$ は*i*番目に観測されたDATAパケットの到着間隔、 $T_{ave.}$ は直近1秒間の平均到着間隔、 $Diff_i$ は*i*番目に観測されたDATAパケットと*i-1*番目とのシーケンス番号の差分を表す。直近の1秒間の平均を用いることで、TCPウィンドウ制御による無通信時間や通信品質の変化など到着間隔が変動することを考慮した。平均到着間隔とシーケンス番号の差分との積をとることで、パケットロス発生時の到着間隔の期待値となる。

各トポロジーでの結果を集計し、順序エラーの発生原因ごとに分類した結果を図5に示す。主縦軸は順序エラーの発生数、第二縦軸は累積頻度を表す。経路変更による順序エラーは91%が期待値以下であり、パケットロスによる順序エラーは87%が期待値以上であった。これは、経路変更による順序エラーは混雑している経路から空いている経路へ切り替わることで発生するため、到着間隔が旧経路での平均値よりも小さくなるためである。また、パケットロスによる順序エラーに関しては、消失したパケットに対する送信処理が消失が発生したノードにおいて行われているため、消失パケット前後のパケットが到着した際にその到着間隔が消失パケットの送信に用いた時間分だけ増加す

る。特に、IEEE802.11 MACは7回のフレーム再送を行うため、パケットロス発生時はバックオフタイムの影響などが加わり、消失パケットの送信に用いる時間は長くなる傾向がある。このことから、経路変更による順序エラーは期待値よりも短くなる傾向が強くなり、パケットロスによる順序エラーは期待値よりも大きくなる傾向が強かった。また、パケットドロップによる順序エラーは、0に集中する結果となった。ただし、母数が327と少なかったこともあり、到着間隔の変動の傾向は不詳である。

## 2.5 考察

予備実験の結果から、経路変更とパケットロスの二つの原因による順序エラーでは、DATA受信時のシーケンス番号の差分と到着間隔にそれぞれ異なる傾向があることが明らかとなった。原因が経路変更の場合はシーケンス番号の差分が大きくなり、到着間隔がパケットロス発生時の期待値以下となる傾向が確認できた。一方、原因がパケットロスの場合は差分が2となる場合が95%以上を占め、到着間隔はパケットロス発生時の期待値以上となる傾向が確認できた。ただし、原因がパケットドロップの場合は発生頻度が小さいため、明確な傾向を確認するに至らなかった。

これらの傾向に基づきシーケンス番号の差分と到着間隔にしきい値を設定し、順序エラーの原因が経路変更かパケットロスかを判定する方法について考える。予備実験の結果から、シーケンス番号の差分にしきい値を設定することで散発的に発生するパケットロスを検知し、到着間隔のしきい値によりバースト的に発生するパケットロスが検知可能であると推測される。表2と表3は、しきい値を設定したときの順序エラー判定精度を表し、経路変更及びパケットロスによる順序エラーのみ含めている。正答率はしきい値によりパ

ケットロスと経路変更を正しく判定できた割合を表し、誤判定率はケットロスが原因の順序エラーを経路変更と誤って判定した割合を表す。

表 2 はシーケンス番号のみで順序エラーの原因を判定し、しきい値を下回るとケットロスと判定し、上回ると経路変更と判定する。しきい値を 2 とすると、全ての原因を経路変更と判定する。そのため、正答率は順序エラー発生回数に対する経路変更を原因とするものの割合であり、誤判定率はケットロスの発生割合を表している。しきい値を 3, 4 へと変化させると、ケットロスの正答率が増加するため、正答率が上がり誤判定率が大きく低下する。これは、図 4 から分かるように、経路変更による順序エラーが発生する環境下では、ケットロスが散発的に発生する傾向があるためである。

表 3 は表 2 に到着間隔のしきい値に式 (1) の  $D_i$  を設定したときの判定精度であり、 $D_i$  が正の値となるとバースト的なケットロスであると判定する。到着間隔にもしきい値を設定することで、正答率をほとんど下げずに誤判定率を下げるができることが分かる。誤判定を行うと、提案手法はケットロスに対する必要な再送制御を抑制してしまうため、誤判定は極力少なくする必要がある。

以上から、経路変更による順序エラーか否かを判断するための条件として、シーケンス番号の差分は 3 パケット分 ( $MSS \times 3$ ) 以上、到着間隔は式 (1) の  $D_i$  の値が負とすることで可能であると考えられる。これは、トポロジーのサイズに依存しないと考えられる。トポロジーのサイズが大きくなると、ケットロス率が増加する傾向にある。シーケンス番号のしきい値を 3 としたとき、トポロジーのサイズが大きくなるにつれて正答率が低下し、誤判定が増加すると予想される。しかし、ケットロス率が高くなっても、連続してケットロスが発生する頻度は非常に小さい。この傾向は、ケットロス率が TCP がまともに通信できるほど高い値 (例えば 10%) となっても、確率的に大きく変化しないと考えられる。また、電波環境等の影響により発生するバーストロスについては、その発生によりシーケンス番号の差分のしきい値を超えることはあるが、これについては到着間隔に基づいた判定により検知する。表 3 から、シーケンス番号のしきい値に基づく判定の誤りを改善していることから、その効果が確認できる。ただし、ケットロス率が 5% 以上と大きくなると、再送と輻輳制御によって TCP 自体

表 2 シーケンス番号のしきい値ごとの判定精度  
Table 2 Determination accuracy for each threshold of sequence number.

しきい値	リング		ランダム 100		ランダム 150	
	正答率	誤判定率	正答率	誤判定率	正答率	誤判定率
2	99.3 %	0.7 %	62.2 %	37.8 %	27.6 %	72.4 %
3	95.3 %	0 %	82.2 %	1.9 %	91.0 %	1.9 %
4	90.3 %	0 %	74.5 %	0.3 %	88.8 %	0.2 %

表 3 到着間隔のしきい値を考慮したシーケンス番号のしきい値ごとの判定精度  
Table 3 Determination accuracy for each threshold of sequence number with reception interval.

しきい値	リング		ランダム 100		ランダム 150	
	正答率	誤判定率	正答率	誤判定率	正答率	誤判定率
2	96.7 %	0 %	89.8 %	4.05 %	89.8 %	7.54 %
3	93 %	0 %	81.6 %	0.32 %	91.7 %	0.13 %
4	89 %	0 %	73.4 %	0.16 %	88.2 %	0.04 %

の通信が阻害されることになる。このときの到着間隔の傾向に関しては、未検証であるため、今後詳細な検証を行っていく必要がある。

### 3. 経路変更に起因する高速再送制御の抑制手法

#### 3.1 提案手法の概要

TCP は、ケット消失以外の原因による順序エラーに対しても再送制御と輻輳制御を行うが、それらは冗長であるという課題がある。この課題の根本的な原因は、受信ノード及び送信ノードが順序エラーが発生した原因や重複 ACK を受信した原因を考慮しないためである。すなわち、これらを解決することで上記課題を解決できると考えられる。ただし、TCP を直接改良することは複雑な設計となるため、提案手法では図 6 に示すように、受信ノードにおいて IP と TCP との間に中間レイヤを挿入する方式を採用する。中間レイヤは、順序エラーの検知とその原因判定を行うモジュール、及び ACK パケットを必要に応じて待機させるモジュールから構成される。これにより、再送を必要としない順序エラーに対する重複 ACK の返信を抑制し、冗長な高速再送制御を抑制する。また、ケットドロップによる順序エラーの場合は、各ノードでのキュー溢れ発生時に TCP ヘッダの予約領域に輻輳発生フラグをセットすることで判定する。

以下、本章では、順序エラーの原因判定手法、冗長な高速再送制御の抑制手法、原因判定手法の誤判定への対策について述べる。

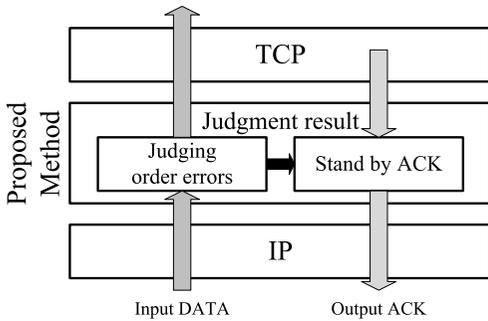


図 6 提案手法の構成  
Fig. 6 Overview of proposed method.

### 3.2 順序エラーの原因判定手法

本手法では、順序エラーの原因が経路変更、パケットロス、パケットドロップのいずれであるかを判定する。2.5の予備実験の考察から経路変更による順序エラーか否かを判断するための条件として、シーケンス番号の差分は3パケット分 ( $MSS \times 3$ ) 以上、到着間隔は2.4で述べた式(1)の  $D_i$  の値が負のときとした。パケットドロップによる順序エラーについては、予備実験の結果からシーケンス番号の差分と到着間隔から判定することは困難であることが分かった。そこで、パケットドロップに関しては、ネットワークに存在する全てのノードでキュー溢れ発生後にエンキューされるDATAパケットに対して、TCPヘッダ内の予約領域に輻輳発生フラグをセットすることで対策する。受信ノードは、輻輳発生フラグが付与されたDATAパケットを受信したときにパケットドロップが原因と判定する。以上にあてはまらない順序エラーについては原因がパケットロスであると判定する。

### 3.3 高速再送制御の抑制手法

送信ノードは重複ACKを3回受信すると高速再送制御を行う。そのため、必要のない高速再送制御を抑制するためには送信ノードが重複ACKを受信しないようにすればよい。そこで、経路変更による順序エラーが発生したときに、経路変更により先行するDATAパケットを追い越して到着したDATAパケットに対するACKパケットを受信ノードにおいて一時的に待機させる。これにより、送信ノードに重複ACKが送信されないため高速再送制御を抑制することができ、自然に順序エラーが回復するのを待つことができる。

提案手法では、ACK待機状態と通常状態の二つの状態を用いてACKの待機が必要か否かを管理する。ACK待機状態は、経路変更による順序エラーの回復待

ち状態であることを表す。高速再送制御の抑制手法は、ACK待機状態時に動作し、重複ACKとなるACKパケットのみ送信を待機する。通常状態では、ACKパケットに関して何も行わない。

ただし、遅れて到着した(追い越された)DATAパケットに対するACKパケットは重複ACKとならないため、その送信は待機させない。また、ACK待機状態で送信されるACKパケットに付与されるSACKオプションは、全て取り除く。これは、送信元ノードが未受信セグメントを示すSACKオプションが付与されたACKパケットを受信すると高速再送制御を行うためである。提案手法では、以上の処理によって、送信ノードに対して順序エラーの発生を隠ぺいしつつ、その目的を達成する。

### 3.4 順序エラー原因判定の誤判定への対策

提案手法は、シーケンス番号の差分と到着間隔を用いて順序エラーの原因を判定する。しかし、パケットロスによる順序エラーを経路変更によるものと誤判定しACK待機状態へ遷移すると、必要な再送を抑制する可能性がある。このとき、送信ウィンドウが枯渇し、最悪の場合RTOが発生する[14]。RTO発生時は輻輳制御によってウィンドウサイズが1に低下するためスループットが低下する。したがって、誤判定時の対策が必要がある。

そこで、提案手法では、ACK待機状態時に、DATAパケットの受信時に毎回タイマ(以下、誤判定検知タイマとする)を設定する。タイマの満了までに順序エラーから回復しなければ判定結果が誤りであったと判断し、ACK待機状態から通常状態へ復帰し、最後に待機したACKパケットとその複製を合計3パケット連続して送信して送信ノードに高速再送制御の実行を促す。これにより、誤判定の場合でも高速再送制御による回復が実現できる。

誤判定検知タイマの設定時刻は、2.4で述べたパケットロス発生時の到着間隔の期待値とする。これは、順序エラーの原因が経路変更であれば、パケットロス発生時の到着間隔の期待値となる時間以内に次のパケットが到着するという予測に基づいている。

## 4. シミュレーション評価

提案手法の有効性を確認するために、ネットワークシミュレータであるQualNet[7]に提案手法を実装し、シミュレーション評価を行った。本評価では、冗長な高速再送制御及び輻輳制御を提案手法により抑制でき

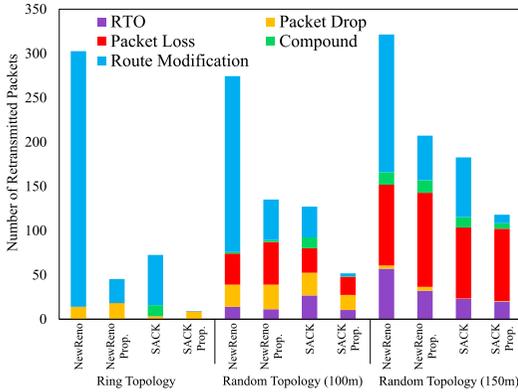


図 7 再送パケット数  
Fig. 7 Number of retransmitted packets.

ることを確認する。評価環境は、2.1 で述べたシミュレーション環境と同一である。

#### 4.1 冗長な高速再送制御の抑制効果

本節では、提案手法による冗長な高速再送制御の抑制効果を、送信ノードが送信した再送パケット数に基づいて評価する。図 7 は、シミュレーションで発生したシナリオごとの再送パケット数を、その発生原因別に積み上げたグラフである。発生原因は、経路変更、パケットロス、パケットドロップと RTO の 4 種類と、これらが複合した原因 (図 7 の Compound) の 5 種類に分類した。縦軸は、評価シナリオを 30 回行ったときの平均再送パケット数である。

提案手法を適用しなかった場合 (図 7 の NewReno と SACK)、経路変更が原因の冗長な再送は、リングトポロジーでは NewReno で 95 %、SACK で 78 %、ランダム 100 トポロジーでは NewReno で 72 %、SACK で 27 %、ランダム 150 トポロジーでは NewReno で 24 %、SACK で 37 %であった。

一方、提案手法適用時は、経路変更による順序エラーが原因の再送パケットが削減されたことを確認できた。提案手法による総再送パケット数の削減率 (図 7 の NewReno と NewReno Prop. 及び SACK と SACK Prop. の比較) は、リングトポロジーでは NewReno で 85 %、SACK で 87 %、ランダム 100 トポロジーでは NewReno で 51 %、SACK で 59 %、ランダム 150 トポロジーでは NewReno で 36 %、SACK で 35 %であった。また、経路変更による冗長な再送パケット数の削減率は、70 %~99 %と高い割合であった。

#### 4.2 冗長な輻輳制御の抑制効果

本節では、提案手法による冗長な輻輳制御の抑制効

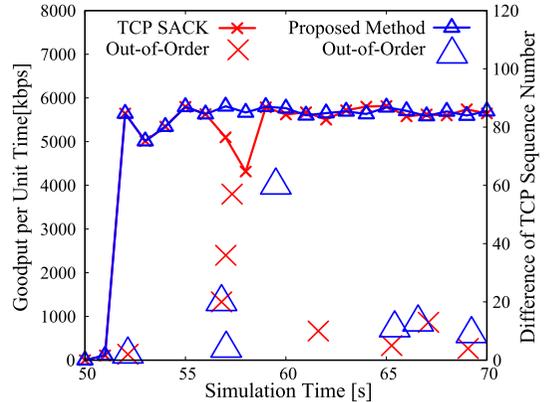


図 8 リングトポロジーでのグッドプットの推移  
Fig. 8 Transition of goodput in ring topology.

果を単位時間当りのグッドプットの推移から評価する。なお、本節では一般的な TCP のデフォルト設定である SACK を比較対象とする。グッドプットはアプリケーションでの受信スループットを意味する。また、本評価はリングトポロジーに限定して述べる。これは、選択可能な経路が二つであることから効果が明確に現れるためである。リングトポロジーで効果があることが示されれば他のトポロジーでも効果があることは自明である。

図 8 にリングトポロジーでの提案手法を適用した場合としない場合のグッドプットの推移を示す。主縦軸は 1 秒間の受信グッドプットを表し (折れ線グラフ)、第二縦軸は順序エラー検出時のシーケンス番号の差分をパケット単位に換算した値である (△と×のプロット)。

シミュレーション開始 52 秒付近で、キュー溢れによる順序エラーが両ケースで発生した。提案手法は、パケットドロップとパケットロス時は再送を妨げないように動作するため、提案手法と SACK 共に高速再送制御と輻輳制御を行ったため、グッドプットが一時的に低下した。

56.8 秒付近では、共に経路変更による 20 パケットの DATA パケットの追越しが発生した。以降、連続して 59.5 秒までの間で合計 100 パケットの追越しが発生した。SACK では、この順序エラーの影響を受け 90 パケットの冗長な再送を行ったため、グッドプットが 23 %低下した。このとき、送信ノードが保持する輻輳ウィンドウサイズは、高速再送制御に伴う輻輳制御によって半分に低下した。半分になった輻輳ウイン

ドサイズは、当該重複 ACK よりも大きいシーケンス番号を保持する ACK パケットを受信しない限り変化しない。すなわち、順序エラーから回復するまで、輻輳ウィンドウサイズは増加しない。

一方、提案手法では、経路変更による順序エラーを検知して、以降に追い越して到着した DATA パケットに対する重複 ACK となる ACK パケットを一時的に送信待機したため、冗長な高速再送制御と輻輳制御が抑制された。そのため、グッドプットの低下が生じず、安定してグッドプットが推移することを確認した。

61 秒以降、SACK では 4 回、提案手法では 3 回の順序エラーが発生している。これらの順序エラーは、経路変更が原因で発生しており、追越し幅が 5 から 13 パケットと 58 秒付近の現象と比較して小さかった。そのため、SACK では、これらの順序エラーにより冗長な高速再送制御と輻輳制御が行われたが、グッドプットに著しい変化は見られなかった。提案手法は、これらの順序エラーに対する冗長な高速再送制御と輻輳制御自体を抑制したため、グッドプットの低下は見られなかった。

以上から、提案手法は冗長な輻輳制御を抑制することでグッドプットの一時的な低下を抑制できることを確認した。また、各トポロジーでの提案手法の適用による平均グッドプットの改善率を表 4 に示す。

### 4.3 考 察

#### 4.3.1 冗長な再送パケットの削減率

シミュレーション評価から、提案手法により冗長な高速再送制御とそれに伴う輻輳制御を抑制できることを確認できた。提案手法を適用することで、経路変更による冗長な再送パケット数を、全ての評価を通じて 70~99 %削減することができた。特に SACK では、提案手法の適用により経路変更による冗長な再送パケット数を 90 %程度削減できることを確認した。NewReno と比較して SACK の冗長な再送パケット数が少ない理由は、NewReno と SACK の再送制御のアルゴリズムの差異によるものである。NewReno は 1 回のウィンドウサイズ分の送信に対して複数のパケットが消失する場合を想定し、冗長度の高いパケ

ト再送アルゴリズムを採用している。一方、SACK は ACK パケットの TCP ヘッダ内にある SACK フィールドに基づき選択的にパケットの再送を行うことができる。しかし、図 7 から明らかなように SACK でも冗長な再送が発生している。このことは、経路変更による順序エラーに対して SACK は有効でないことを意味する。これに対して、提案手法を適用すると経路変更起因する順序エラーによる再送を抑制することができるため、全体に占める冗長な再送パケットの割合を 5 %程度にまで削減することが実現できた。全ての冗長な再送を抑制できなかったのは、次節で述べる順序エラーの原因判定の誤判定によるものである。

また、提案手法は冗長な高速再送制御と輻輳制御を抑制したことで不要な帯域の浪費を防ぎ、平均グッドプットの向上に貢献できた。しかし、ランダム 150 トポロジーかつ SACK 使用時のケースについてのみ、平均グッドプットが低下した。その原因は、パケットロス率が平均 0.5 %と高かったことにある。パケットロスに対して提案手法は、再送が必要なため、再送制御を TCP に委任している。パケットロス発生時は TCP による再送だけでなく輻輳制御も行われ、輻輳ウィンドウサイズの低下、すなわちグッドプットが低下する。そのため、高パケットロス環境では、経路変更による影響よりもパケットロスの方が影響が大きくなるため、提案手法の効果が表れにくい。また、提案手法は、正答率が 92.7 %、誤判定率が 0.2 %であり、RTO の増加も確認されなかったことから、グッドプット低下の原因ではないと考えられる。

#### 4.3.2 原因判定の精度としきい値の妥当性

提案手法による順序エラーの原因判定の精度は 91.4 %であった。特にパケットロスの正答率は 99.6 %であった。提案手法による誤判定のパターンは以下に挙げる四つが考えられる。

- (1) 経路変更と判定したがパケットロスであった。
- (2) 経路変更と判定したがパケットドロップであった。

- (3) パケットロスと判定したが経路変更であった。

- (4) 判定結果にかかわらず複数の原因が複合した。

(1) の場合は、本来行われるべき再送を抑制することになる。しかし、(1) は全体の 0.2 %と小さい割合であった。更に、提案手法では (1) のような場合に対して 3.4 で述べた誤判定検知タイムを用いて対策を行うことで、その影響を最小限に抑えている。(2) は輻輳発生フラグによって判定を行うため誤判定は発

表 4 提案手法によるグッドプットの改善率

Table 4 Improvement rate of Goodput by proposed method.

	リング	ランダム 100	ランダム 150
NewReno	3.69 %	0.9 %	5.17 %
SACK	0.28 %	1.54 %	-1.79 %

生しない。シミュレーション結果からも (2) が発生しなかったことを確認している。(3) の場合は、重複 ACK を抑制しないため通常の TCP と同じ処理が行われる。すなわち、冗長な再送が発生する可能性がある。ただし、その割合は全体の 8.4% であった。更に、誤判定の原因がシーケンス番号のしきい値を超えなかったことにある場合は、高速再送制御が起動する前に順序エラーが修正されることが多い。このようなケースが (3) の中で 89% であったことから、(3) による影響は小さいと考えられる。

提案手法は、(4) を判断することができない。この場合、必ずパケットの消失が発生しているため、消失した DATA パケットの再送が必須である。しかし、順序エラー検出時にどのパケットが消失したのかを判断できない。提案手法では順序エラーの判定結果により (1) と (3) に処理が分かれる。すなわち、どのような判定結果であっても提案手法が必要な再送制御を阻害することはない。また、(4) のパターンは、誤判定の中でも 1.6% と割合が低く、その影響は小さいと考えられる。

以上から、提案手法における順序エラーの原因が経路変更によるものであると正しく判定される精度は 91.4%、誤判定は 0.2%、検出もれは 8.4% であった。順序エラーの原因判定のしきい値は、シミュレーション環境に依存しにくいと考えられる。これは次の理由による。パケットドロップに関しては提案手法では輻輳検知フラグを用いて検知するため、シミュレーション環境に依存しない。また、パケットロスによる順序エラーのシーケンス番号の傾向に関してはパケットロス率に依存するが、パケットの衝突等により一意に定められるものではない。本シミュレーションにおけるネットワーク全体のパケットロス率はリングトポロジーで 0.02%、ランダム 100 トポロジーで 0.13%、ランダム 150 トポロジーで 0.56% であり、最悪なケースでは 1% を越えるものがあつた。パケットロス率は干渉となるトラフィックが複数ある環境や中継回数が多い環境ほど高くなるが、提案手法はトポロジーが小さくパケットロス率が低い環境でも、トポロジーが大きくパケットロス率が高い環境でも冗長な再送を抑制するという目的は達成していることから、提案手法が目的とする冗長な再送制御の抑制という観点では今回のシミュレーションで用いた環境以外でも有効だと考えられる。更に、経路変更の発生頻度はルーティングプロトコルに依存するが、無線メッシュネットワークで

は、hop-by-hop のリンクステート型ルーティングプロトコルが最適とされている [1], [2]。ディスタンスベクタ型ルーティングプロトコルを用いることで、経路変更による順序エラーの発生を抑制できるが、リンクステート型よりも通信性能が悪くなることがいわれている。実際に AODV を用いて 2.1 のシミュレーション環境と同様の環境で評価を行ったところ、経路変更による順序エラーは発生しなかったが、スループットが 36% 低下することを確認している。また、トラフィックデマンドにより経路変更による順序エラーの発生頻度は変動すると考えられる。昨今の WEB アクセスでは画像や動画といったメディアの転送が多く、FTP のようなトラフィックが発生すると考えられる。無線メッシュネットワークの主な用途がインターネットアクセスであること考慮すると、WEB アクセスによるトラフィックが支配的になると考えられるが、そのトラフィック特性は今回の FTP に近いものになると考えており、この点についても一般的な環境と大きな乖離のない評価ができていますと考えられる。また、LINE などのメッセージ系アプリケーションは、トラフィック量、パケット送信間隔が小さいことから経路変更による順序エラーは発生しにくいと考えられる。以上から、一般的な無線メッシュネットワークにおいて、二つのしきい値を用いる提案手法は経路変更に起因する冗長な高速再送制御の抑制に有効だと考えられる。

#### 4.3.3 誤判定検知タイマの効果

提案手法では、4.3.2 で挙げた 4 パターンの誤判定のうち (1) と (4) で誤判定検知タイマを用いて対処している。ランダム 150 トポロジーではパケットロス率が高いため、誤判定検知タイマが 1 回のシミュレーション当り 2 回満了していた。ランダム 150 トポロジーで誤判定検知タイマを用いず提案手法の評価を行ったところ、誤判定検知タイマ有りと比較して、RTO 発生数が平均 11 回から 13 回に増加、総再送数は 33% 増加、平均グッドプット 3% 減少と、提案手法の効果が減少していることが分かった。また、リングトポロジーでは誤判定検知タイマは満了せず、ランダム 100 トポロジーでは平均 0.5 回満了していた。ランダム 100 トポロジーでも、誤判定検知タイマを用いなかった場合に提案手法の効果が減少していることを確認した。以上から、誤判定検知タイマは、順序エラーの原因判定での誤判定による影響を抑えることを確認した。

#### 4.3.4 実環境への適用可能性

近年の TCP は、Linux では CUBIC, Windows では Compound TCP が用いられている。これらの輻輳制御アルゴリズムでも、パケットの消失検知を重複 ACK を用いて行っている。提案手法は、高速再送制御のトリガである重複 ACK を抑制するため、輻輳制御アルゴリズムに依存せず、その効果を発揮することができる。

実機実装については、提案手法は 3.1 で述べたように IP と TCP との間の中間レイヤに実装することを前提としている。Linux では、netfilter [15] と呼ばれる IP 層の複数ポイントでパケットの操作が可能なフレームワークがあり、LKM (Loadable Kernel Module) としてカーネルに機能を追加することができる。Windows でも NDIS ドライバ [16] が netfilter と同様の機構を提供している。これらの機構を利用することで、実機への実装をカーネルのリビルドをせずに実装することができる。我々は、既に提案手法のプロトタイプ (パケットドロップに対して未対策) を LKM として実装し Linux で評価を行っている [6]。評価の結果、SACK 付き CUBIC において冗長な再送パケット数を 87% 削減し、Goodput は約 1% 程度向上できることを確認し、実環境においても提案手法が有効であることを確認している。

## 5. 関連研究

TCP に順序エラー耐性をもたせる研究は幾つか存在する。RR-TCP [17], RD-TCP [18], TCP-PR [19] は有線環境でのマルチパスルーティングや負荷分散による順序エラーを想定している。これらの研究は、重複 ACK の受信処理の改良による高速再送制御の発生タイミングの変更や、輻輳制御アルゴリズムの変更により冗長な輻輳制御を抑制することで通信性能の低下を抑制する。RD-TCP は、提案手法と同様に全てのノードに対して輻輳検知機構を適用し、パケットドロップを明示的に通知することでマルチパスルーティングによる順序エラーとパケットドロップとの判定を行う。これらは、無線リンクにおけるパケットロスを想定していないため、提案手法と同じ無線メッシュネットワークに適用しても効果が小さいと考えられる。

TCP-DOOR [20] は、無線マルチホップネットワークを想定し、経路変更により発生する冗長な輻輳制御の抑制を目的としている。順序エラーの検知は DATA パケット及び ACK パケットに対して送信順 (再送パ

ケットを含む) に番号を付与することで実現する。順序エラー検出時は一時的に輻輳制御の実行を止めて TCP の各種状態 (RTO タイマや輻輳ウインドウサイズなど) の保存を行い、順序エラー回復時には保存していた TCP の各種状態を用いて順序エラー検出前の状態に戻すことで高速な輻輳回避状態からの復帰を実現する。TCP-DOOR は、経路変更による順序エラーを対象としつつも、それによる冗長な再送を抑制することを目的とはしておらず、効果的な輻輳制御の実現を目的としている。また、順序エラー検出の方法は、その原因をパケットロスなのか、経路変更なのかを判定するものではない。したがって、TCP-DOOR の順序エラー検出方法は、冗長な再送制御には不十分である。

TCP-DCR [21] は、無線と有線が混在する環境を想定し、高速再送制御を RTT 分待機することで輻輳が原因でない順序エラーに対して不必要な輻輳制御を抑制する。TCP-DCR は無線リンクでのパケットロス、有線リンクでの順序エラー及び輻輳によるパケットドロップを考慮し、パケットロスは無線 LAN の基地局で再送することで送信ノードでの再送を抑制する。この基地局による再送を考慮し、TCP-DCR の送信ノードでは高速再送制御時に RTT 相当の待機時間を挿入する。しかし、TCP-DCR は、TCP の適用だけでなく中継するルータにも改良が必要であるため導入コストが高い。また、提案手法とは異なり、順序エラー発生時の輻輳制御を対象とした研究である。

このように、順序エラー耐性をもつ TCP は、順序エラー発生時によって行われる輻輳制御を対象とし、輻輳ウインドウサイズを大きく維持してスループットを向上させることを目的としている。一方、提案手法は、経路変更によって発生する順序エラーに焦点を絞り、冗長な再送を抑制することを目的としている。また、その対象は高速再送制御である。そのため、提案手法では、順序エラー発生時の原因を判別しているが、既存研究では、順序エラーの発生原因の判定までは行っていない。このように、提案手法は、その目的と制御対象及び原因判定まで行う機能を有する点で、既存研究と異なる。逆に、提案手法はパケットロス発生時の輻輳制御について対策を行っていない。そのため、高パケットロス環境下では、冗長な再送制御の抑制効果はあるが、スループットについては、TCP と同等に悪化する。この点については、既存の無線パケットロスに対応した輻輳制御アルゴリズムと併用するこ

とで対応することができる。また、提案手法の順序エラー判定手法は、汎用的なものであるため、輻輳制御アルゴリズムに適用するなどの応用が期待できる。

## 6. む す び

本論文では、無線メッシュネットワークにおける経路変更によって発生する冗長な高速再送制御を抑制する手法について述べた。TCP は、順序エラーの原因に関係なく重複 ACK を 3 パケット受信すると再送制御を行うため、冗長な再送及び輻輳制御が課題であった。この課題に対して、提案手法では、DATA パケットのシーケンス番号の差分と到着間隔、TCP ヘッダに追加した輻輳発生フラグの三つを用いて順序エラーの原因を判定し、経路変更であると判定すると重複 ACK となる ACK パケットの送信を待機することで高速再送制御を抑制する。シミュレーション評価の結果、二つのしきい値とフラグを用いることで 91% 正しく順序エラーの原因を判定し、冗長な再送パケット数は全ての評価を通じて 70~99 % 削減することが確認できた。また、提案手法は、冗長な高速再送制御を抑制したことで不要な帯域の浪費を防ぎ、平均スループットを 1~5 % の向上を確認した。

今後は順序エラーの判定結果を送信ノードへ伝え、パケットロスが原因の場合に輻輳制御を抑制する手法を検討する予定である。

## 文 献

- [1] Y. Yaling, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," Proc. IEEE Workshop on WiMesh '05, 2005.
- [2] M.E.M. Campista, P.M. Esposito, I.M. Moraes, L.H.M.K. Costa, O.C.M.B. Duarte, D.G. Passos, C.V.N. deAlbuquerque, D.C.M. Saade, and M.G. Rubinstein, "Routing metrics and protocols for wireless mesh networks," IEEE Network, vol.22, no.1, pp.6-12, Jan. 2008.
- [3] E. Gelenbe and M. Gellman, "Can routing oscillations be good? the benefits of route-switching in self-aware networks," Proc. IEEE MASCOTS '07, pp.343-352, Oct. 2007.
- [4] Z. Zaidit, T.Y. Tan, and Y. Cheng, "ETX could result in lower throughput," Proc. IEEE ICCCN '09, pp.1-6, Aug. 2009.
- [5] K.-C. Leung, V.-K. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (TCP): Problems, solutions, and challenges," IEEE Trans. Parall. Distrib. Syst., vol.18, no.4, pp.522-535, April 2007.
- [6] S. Aketa, E. Takimoto, Y. Otsuki, S. Saito, E.W. Cooper, and K. Mouri, "Suppressing redundant TCP retransmissions in wireless mesh networks," Proc. IAENG IMECS '15, vol.2, pp.556-561, 2015.
- [7] QualNet, <http://web.scalable-networks.com/content/qualnet> Nov. 2015.
- [8] D.S.J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," Proc. ACM MobiCom '03, pp.134-146, 2003.
- [9] T. Clausen, C. Dearlove, P. Jacquet, and U. Herberg, "The optimized link state routing protocol version 2," RFC 7181, 2014.
- [10] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The newreno modification to TCP's fast recovery algorithm," RFC 6582, 2012.
- [11] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgment options," RFC 2018, 1996.
- [12] V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," RFC 1323, 1992.
- [13] J. Nagle, "Congestion control in IP/TCP internet networks," RFC 896, 1984.
- [14] 明田修平, 大月勇人, 毛利公一, 瀧本栄二, "経路変更に起因する冗長な TCP 再送制御の抑制手法の改善," 信学技報, MoNA2013-46, Nov. 2013.
- [15] The netfilter.org project, <http://www.netfilter.org/> March 2016.
- [16] NDIS Developer's Reference, <http://www.ndis.com/> March 2016.
- [17] M. Zhang, B. Karp, S. Floyd, and L. Peterson, "RR-TCP: A reordering-robust TCP with DSACK," Proc. IEEE ICNP '03, pp.95-106, Nov. 2003.
- [18] A. Sathiaselvan and T. Radzik, "RD-TCP: Reorder detecting TCP," Proc. IEEE HSNMC '03, pp.471-480, 2003.
- [19] S. Bohacek, J.P. Hespanha, J. Lee, C. Lim, and K. Obraczka, "A new TCP for persistent packet reordering," IEEE/ACM Trans. Netw., vol.14, no.2, pp.369-382, April 2006.
- [20] F. Wang and Y. Zhang, "Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response," Proc. ACM MobiHoc '02, pp.217-225, 2002.
- [21] S. Bhandarkar, N.E. Sadry, A.L.N. Reddy, and N.F. Vaidya, "TCP-DCR: A novel protocol for tolerating wireless channel errors," IEEE Trans. Mobile Comput., vol.4, no.5, pp.517-529, 2005.

(平成 27 年 12 月 3 日受付, 28 年 3 月 18 日再受付, 4 月 13 日早期公開)



明田 修平

2013年立命館大学情報理工学部情報システム学科卒業，2015年同大学大学院情報理工学研究科博士前期課程情報理工学専攻修了．同年同大学大学院情報理工学研究科博士後期課程情報理工学専攻に入学，現在に至る．コンピュータネットワークに関する研究に従事．情報処理学会会員．



瀧本 栄二

1999年立命館大学理工学部情報学科卒業，2001年同大学大学院理工学研究科博士前期課程修了，2005年同研究科博士後期課程単位取得退学，同年（株）ATR 適応コミュニケーション研究所専任研究員，2010年立命館大学情報理工学部情報システム学科助手，現在に至る．主にシステムソフトウェア，無線通信に関する研究に従事．博士（工学）．情報処理学会会員．



大月 勇人

2011年立命館大学情報理工学部情報システム学科卒業，2013年同大学大学院理工学研究科博士前期課程情報理工学専攻修了．2016年同大学大学院情報理工学研究科博士後期課程情報理工学専攻修了，同年日本電信電話株式会社入社．現在，NTTセキュアプラットフォーム研究所勤務．マルウェア解析技術に関する研究に従事．博士（工学）．情報処理学会会員．



齋藤 彰一

1993年立命館大学理工学部情報工学科卒業．1995年同大学大学院博士前期課程修了．1998年同大学大学院博士後期課程単位習得中退．同年和歌山大学システム工学部情報通信システム学科助手．2003年同講師，2005年同助教授．2006年名古屋工業大学大学院助教授，2007年同准教授，2016年同教授，現在に至る．オペレーティングシステム，インターネット，セキュリティなどの研究に従事．博士（工学）．情報処理学会，ACM，IEEE-CS 各会員．



毛利 公一（正員）

1994年立命館大学理工学部情報工学科卒業，1996年同大学大学院理工学研究科修士課程情報システム学専攻修了，1999年同研究科博士課程後期課程総合理工学専攻修了．同年東京農工大学工学部情報コミュニケーション工学科助手，2002年立命館大学理工学部情報学科講師，2004年同大学情報理工学部情報システム学科講師，2008年同准教授，2014年同教授となり，現在に至る．博士（工学）．オペレーティングシステム，仮想化技術，コンピュータセキュリティ等の研究に従事．情報処理学会，ACM，IEEE-CS，USENIX 各会員．