

特別研究報告

題目

無線ネットワーク環境における TCP の性能改善方式の提案

指導教官

村田 正幸 教授

報告者

加藤 雄一郎

平成 13 年 2 月 22 日

大阪大学 基礎工学部 情報科学科

無線ネットワーク環境における
TCP の性能改善方式の提案

加藤 雄一郎

内容梗概

伝送誤り率が低い有線ネットワークにおいては、TCP セグメントの損失発生は、輻輳が発生していると解釈され、TCP はセグメントの損失を検出すると、輻輳制御を行なう機構を備えている。一方、無線回線では、有線回線とは異なって雑音や干渉などの影響を受けやすく、データ伝送誤り率が高いために、伝送誤りによるセグメント損失も発生する。このような無線回線上でのセグメント損失は、ネットワークの輻輳状態とは無関係に発生するため、輻輳制御を行なう必要はない。しかしながら、TCP はセグメント損失の原因を区別することができないため、一律に輻輳制御を行うことによってスループットが非常に低下するという問題が発生する。

そこで、本報告では、無線ネットワーク環境において、無線回線上でのセグメント損失に起因する TCP の性能劣化を改善するための方式を提案する。本方式では、モバイル端末が伝送誤りを検出した場合には、それを送信側に知らせる。その結果、送信ホストにおける輻輳制御を回避することができ、スループットを向上させることができる。無線回線上の誤り発生情報の伝達は、TCP Selective Acknowledgement (SACK) 方式のオプションを拡張することによって行い、基地局などを含む中間ノードでの制御を必要としないという特徴がある。本報告では、シミュレーションによって従来の TCP と提案方式の性能比較を行なった結果、スループットの向上ができることを示した。また、無線回線上の誤り発生率が非常に高い場合に、FEC (Forward Error Correction) を適用した場合でも本方式による再送を行なうことで性能劣化をより改善できることを明らかにする。

主な用語

TCP (Transmission Control Protocol)、無線ネットワーク環境、伝送誤り、輻輳制御、TCP SACK 方式 (TCP Selective Acknowledgement)、FEC (Forward Error Correction)

目次

1	はじめに	5
2	無線ネットワーク環境における TCP	8
2.1	TCP の誤り制御と輻輳制御	8
2.1.1	TCP の誤り制御	8
2.1.2	TCP の輻輳制御	11
2.2	無線ネットワーク環境での TCP の問題	14
2.2.1	無線ネットワーク環境	14
2.2.2	無線環境下の TCP に関する従来の研究	15
3	提案方式の概要	17
3.1	TCP SACK 方式	17
3.2	提案方式	20
4	シミュレーションによる評価	25
4.1	ネットワークモデル	25
4.2	提案方式の評価	26
4.2.1	TCP スループット	26
4.2.2	輻輳ウィンドウサイズの変化	28
4.3	FEC の適用	30
4.3.1	セグメント誤り率の算出	30
4.3.2	シミュレーションによる評価	32
5	おわりに	37
	謝辞	38
	参考文献	39

目 次

1	無線ネットワーク環境の例	6
2	タイムアウトによるセグメント損失の検出と再送	8
3	累積確認応答	10
4	重複 ACK によるセグメント損失の検出と早期再送	10
5	ウィンドウフロー制御	11
6	スロースタートフェーズと輻輳回避フェーズ	12
7	TCP Reno 方式の輻輳ウィンドウサイズの変化	13
8	SACK オプションフィールドの具体例	18
9	提案方式における ACK の種類と再送方式	22
10	提案方式オプションフィールドの具体例	23
11	対象とするネットワークの構成	25
12	TCP スルーブット	27
13	輻輳ウィンドウサイズの変化 (セグメントサイズ: 512 bytes)	29
14	リードソロモン符号のブロック構成	30
15	QPSK 変調でのビット誤り率特性	31
16	セグメントとフレームの関係	32
17	RS 符号を適用したときの TCP スルーブット	33
18	セグメント誤り率	34
19	RS(127,117) を適用したときの TCP スルーブット	35
20	RS(127,107) を適用したときの TCP スルーブット	36

1 はじめに

近年、無線ネットワークの急速な発展とインターネットの普及により、携帯電話をはじめとするモバイル端末からインターネットに接続するユーザ数が急激に増加している。次世代の移动通信環境として開発が進められている IMT-2000 (International Mobile Telecommunications-2000) においても、従来の音声だけでなくデータ通信が重視されており、今後ますます無線ネットワークでのデータ通信が広がっていくものと考えられる (図 1)。現在、インターネット上で利用されている電子メールやファイル転送などのアプリケーションは、下位層のプロトコルとして TCP (Transmission Control Protocol) [1] を用いるため、無線回線を介した TCP の通信が増加していくことが予測できる。

TCP は、有線回線で構成されたインターネット上で開発が行なわれてきたプロトコルである。有線回線は通信品質が高く安定しており、TCP の情報伝送単位であるセグメントの損失はほとんど発生しない。つまり有線ネットワークにおいてセグメントの損失が発生するのは、ネットワークに対する負荷が過剰になっており、輻輳が発生しているためと考えることができる。そのため、TCP はセグメントの損失に基づいて輻輳制御を行なう機構を備えている。すなわち、送信側がセグメントの損失を検知すると、ウィンドウサイズを縮小することによってネットワークに対する送出量を抑えることにより、輻輳状態を回避しようとする。一方、無線回線は、有線回線とは異なって雑音や干渉などの影響を受けやすく誤り発生率が非常に高いために、伝送誤りによるセグメントの損失が発生する。このような無線回線上でのセグメントの損失は、ネットワークの輻輳状態とは無関係に発生するため、輻輳制御を行なう必要はない。しかしながら、TCP は、セグメントの損失が輻輳によるものか、伝送誤りによるかを区別することができないため、一律に輻輳制御を行なってデータの送出量を抑えてしまう。その結果として、無線ネットワーク環境下ではスループットが非常に低下するという問題が発生する。

このような無線回線における TCP の性能劣化の問題に着目し、それを回避するための方法がいくつか提案されている [2-4]。それらの方法は、TCP コネクションの中間に位置し、無線端末と直接通信を行なうことのできる基地局に対して何らかの工夫を行なうという手段に基づいている。ところが、このような中間ノードで TCP の制御を行う方式は、コネク

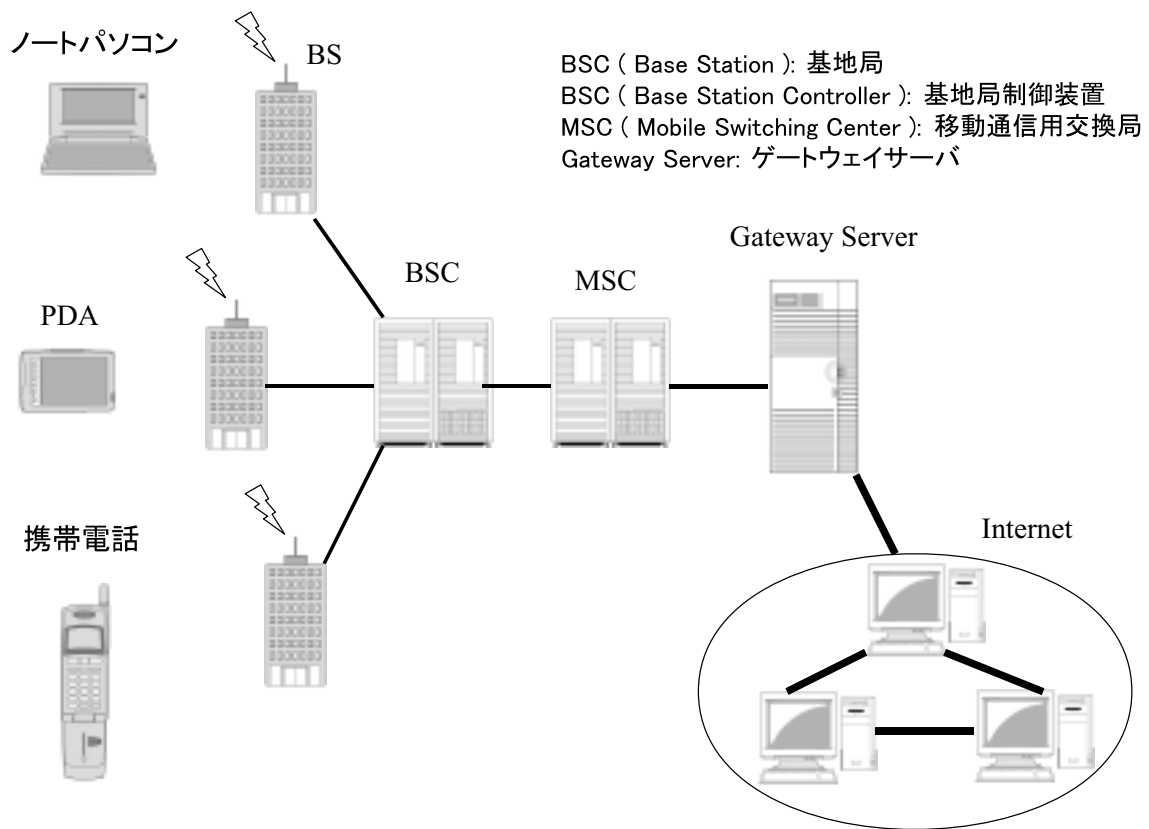


図 1: 無線ネットワーク環境の例

ション数が多い場合のバッファ管理の問題や暗号化によってパケットモニターができない場合など、実現する際に解決すべき問題が多く、現実には実装が困難であると考えられる。

そこで、本報告では、無線ネットワーク環境において、無線回線上でのセグメント損失に起因する TCP の性能劣化を改善するための方式を提案する (ELNSACK: Explicit Loss Notification by SACK enhancement)。本方式は、有線ネットワーク側のホストから無線端末へのデータの転送を想定している。無線端末において伝送誤りが検出された場合には、それを送信側に知らせることにより、セグメント損失が発生しても、送信ホストにおける輻輳制御を回避することができ、スループットの向上が期待できる。無線回線上の誤り発生情報の伝達は、TCP Selective Acknowledgement (SACK) 方式 [5, 6] のオプションを拡張することによって行なう。本方式において変更を加えるのは送受信ホストの TCP 機構のみであり、基地局などの中間ノードにおける制御は必要でないため、従来の研究で提案されている他の方式に比べて実現性が高い。本報告では、シミュレーションによって、従来の TCP と提案方式のスループットの比較を行い、本方式において無線ネットワーク環境下での性能劣化を改善することを示す。また、無線回線伝送品質が低く、誤り発生率が非常に高い場合には、一般にセグメントの損失を抑えるために FEC (Forward Error Correction) を用いるが、そのような場合にも本方式による再送を行なうことで、性能をより改善できることを明らかにする。

本報告の構成は以下のとおりである。まず、2 章では、無線ネットワーク環境の問題、および関連研究の概要を説明する。3 章では、TCP SACK 方式、および提案方式を説明する。続いて 4 章では、シミュレーションによる提案方式、および FEC を用いた手法の評価を行なう。最後に 5 章では、本報告のまとめと今後の課題について述べる。

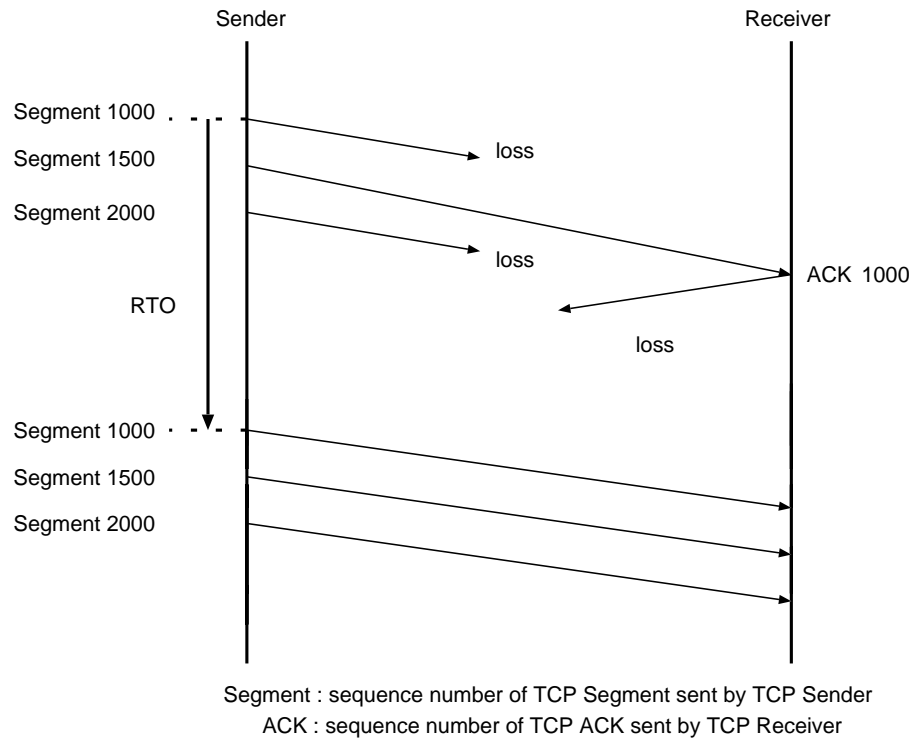


図 2: タイムアウトによるセグメント損失の検出と再送

2 無線ネットワーク環境における TCP

2.1 TCP の誤り制御と輻輳制御

TCP [1] はトランスポート層のプロトコルであり、上位のセッション層からの要求により、送信、受信ホスト間の接続の設定や開放を行なう。また、TCP は信頼性の高い通信を提供するために、誤り制御や輻輳制御も行なっている。以下では、TCP の誤り制御および輻輳制御について説明を行なう。

2.1.1 TCP の誤り制御

TCP では、タイムアウトと重複 ACK 受信という二通りの方法でセグメントの損失を検出する。まず、図 2 によって、タイムアウトによるセグメント損失の検出を示す。図 2 の送受信に記された番号はセグメントのシーケンス番号を表し、受信側の番号は ACK のシーケン

ス番号を表している。送信側は、セグメントを送るたびに、毎回タイマーをスタートさせて ACK (確認応答) を待つ。受信側はセグメントを受信した場合には ACK を返す。図 2 に示すようにあるセグメントが損失すると ACK は戻ってこないことになる。そこで、ACK が戻る前にタイマが一定時間を越えると、送信側はセグメントが損失したものと判断して再送を行なう。これがタイムアウトによるセグメント損失の検出である。しかし、インターネットでは同じ送受信ホスト間でも負荷の変動によってセグメントが受信側へ到達するまでの時間は変動する。したがって、タイムアウト時間を設定するにはこの点を考慮に入れる必要がある。一般にタイムアウト時間 (*RTO*) は往復遅延時間 (*RTT*) を見積もってそれをもとに決定している。

このようにセグメント損失の検出をタイムアウトによって行った場合、損失が発生してから長期間に渡り送信が停止されることになる。ネットワークの高速化に伴って、この停止時間は TCP の性能を著しく劣化させる原因になる。そこで TCP では、タイムアウト以外に、重複 ACK によってセグメント損失を検出する機能を備えている。まず、TCP では図 3 に示すように、正しい順序で連続したセグメントをどこまで受け取ったかを知らせるために累積確認応答を用いている。ここでは、セグメントのサイズを 500 バイトとしており、シーケンス番号 500 番のセグメントが到着した場合は、シーケンス番号 1000 番の ACK を返す。ところが図 4 のように、1000 番のセグメントが失われることによって、500 番のセグメントの次に到着したセグメントが 1500 番であった場合、1500 番のセグメントに対しては累積確認応答により 1000 番の ACK を返すこととなる。次に到着した 2000 番のセグメントに対しても同様に 1000 番の ACK を返す。このように累積確認応答により発生した同じシーケンス番号の ACK を重複 ACK (duplicate ACK) という。重複 ACK が発生するのは、セグメントが途中の経路の変更によって到着順序が逆転するときや、あるセグメントが損失して、それ以降のセグメントが到着したときである。順序の逆転による重複 ACK である場合のことも考慮して、3 個以上の重複 ACK が連続して戻ってきた場合に、送信側はセグメントが損失したと判断して、タイムアウトを待たずにセグメントの再送を行なう。これを早期再送 (fast retransmit) 方式と呼ぶ。図 4 に示すように、重複 ACK を用いることにより、タイムアウトよりも短時間でセグメントの損失を検出でき、TCP の性能が改善される。

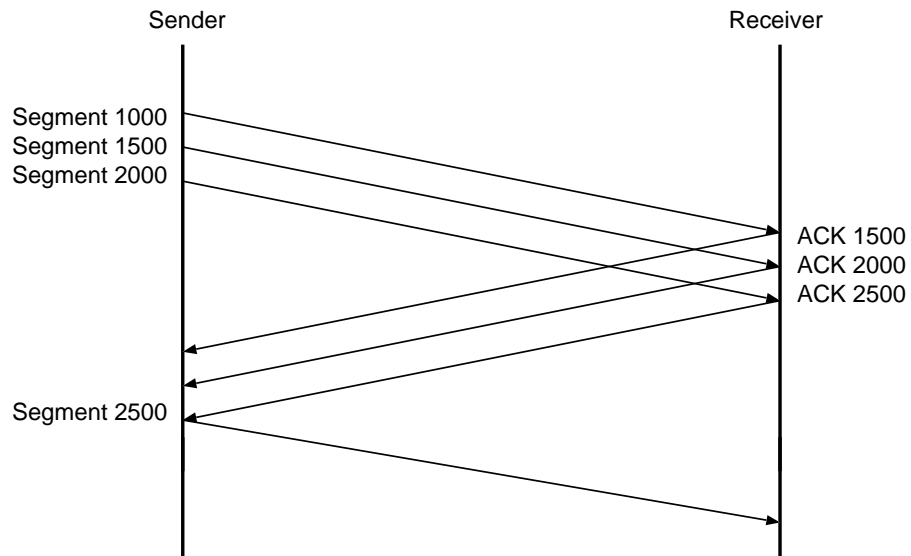


図 3: 累積確認応答

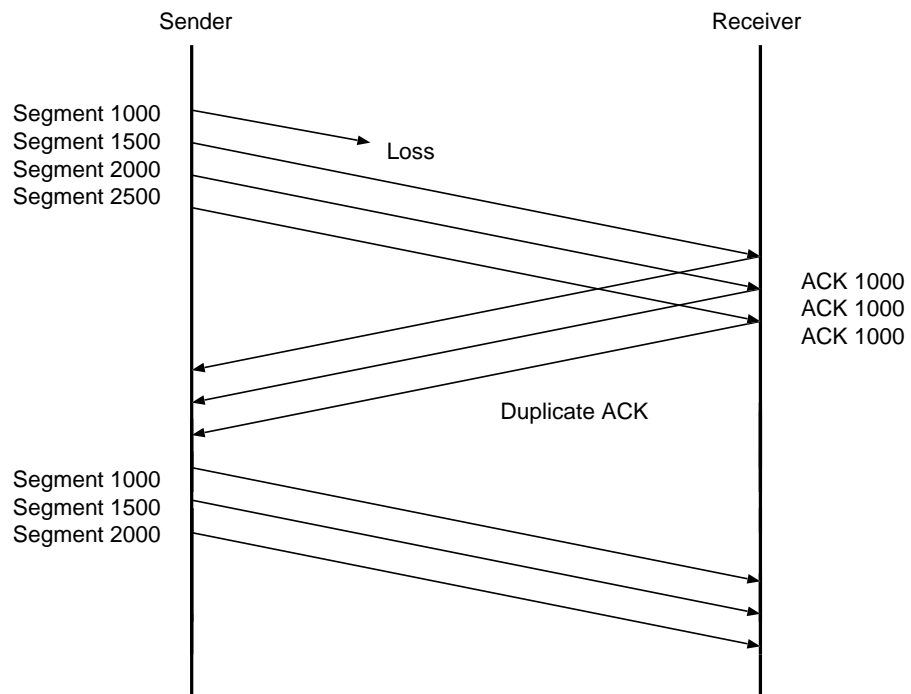


図 4: 重複 ACK によるセグメント損失の検出と早期再送

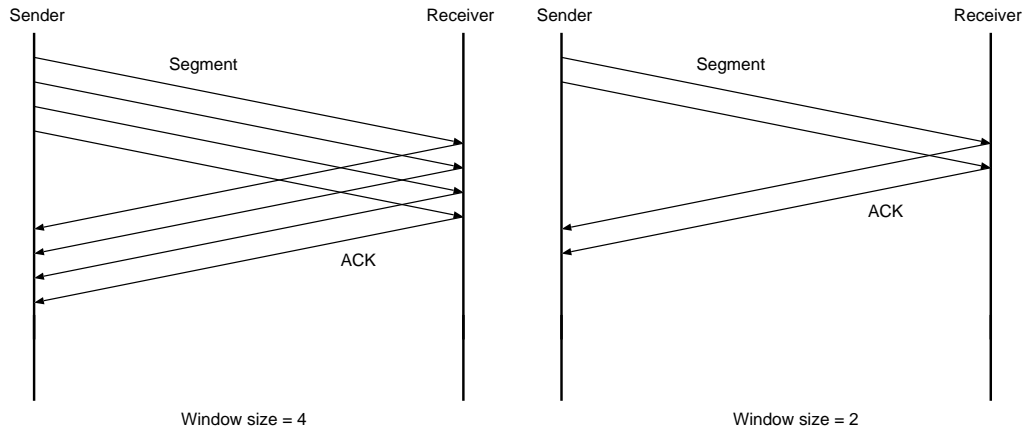


図 5: ウィンドウフロー制御

セグメントの損失を検出すると送信側は再送を行なうが、再送方式として最も多くの TCP が採用している ARQ (Automatic Repeat reQuest) は Go-back-N 方式である。すなわち、 i 番目で誤りを検出したとき、再びそれが正しく受信されるまでの間に受け取った $i + 1$ 番目以降のセグメントはたとえそれらが正しく受信されたとしても破棄する方式である。図 2、図 4 はいずれもシーケンス番号 1000 番のセグメントが損失したために、再びシーケンス番号 1000 のセグメントから再送している。

2.1.2 TCP の輻輳制御

TCP が採用している輻輳制御では、セグメント単位のウィンドウフロー制御 (図 5) を行なう。ウィンドウサイズは受信側が設定する最大セグメント数を示した広告ウィンドウサイズ以内で、動的に変更される。タイムアウトや重複 ACK の受信によってセグメントの損失を検出すると、ウィンドウサイズを動的に調整することによって輻輳状態を回避しようとする。その変更には 2 通りの方法があり、スロースタートフェーズ (slow start phase) および輻輳回避フェーズ (congestion avoidance phase) でそれぞれで、ウィンドウサイズの変更方法が異なる。また、輻輳はセグメントの損失によって検出するが、セグメントの損失を重複 ACK もしくはタイムアウトによって検出した際にウィンドウサイズを動的に調整することによって輻輳状態を緩和する。

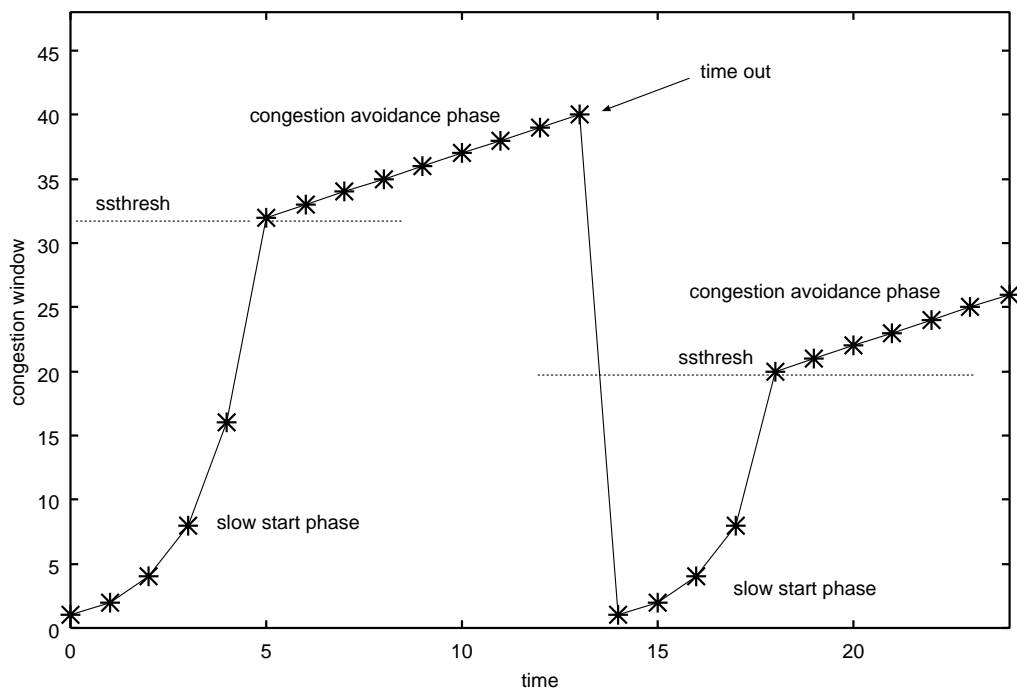


図 6: スロースタートフェーズと輻輳回避フェーズ

スロースタートフェーズは送信開始およびセグメント損失後の送信再開後に行なわれる輻輳制御である。まず、初期状態として輻輳ウィンドウサイズ $cwnd$ を 1 セグメントに設定する。 $cwnd$ はセグメント単位である。新しい ACK を受け取るごとに 1 セグメント分 $cwnd$ を増加させる。このような手法により、スロースタートフェーズでは、 $cwnd$ は $1RTT$ ごとに 2 倍に増加する。 $cwnd$ がしきい値 $ssthresh$ に達すると、スロースタートフェーズから輻輳回避フェーズに移行する。

輻輳回避フェーズにおいては、 $cwnd$ は利用可能な帯域に近づいたと考えられ、比較的緩やかなウィンドウ増加を行なう。すなわち新しい ACK を受け取るごとに $cwnd$ を $1/cwnd$ ずつ増加する。したがって、 $1RTT$ の間では、セグメントの損失が発生しなければ、 $cwnd$ 個の ACK が戻ってくるため、 $1RTT$ に 1 セグメント分ウィンドウサイズを増加することになる。図 6 にスロースタートフェーズと輻輳回避フェーズの動作を示す。

セグメント損失を検出する方法のうち、タイムアウトによる損失検出では、ACK が返って来なかったという観点から、ネットワークが過負荷状態になっており、送信するセグメン

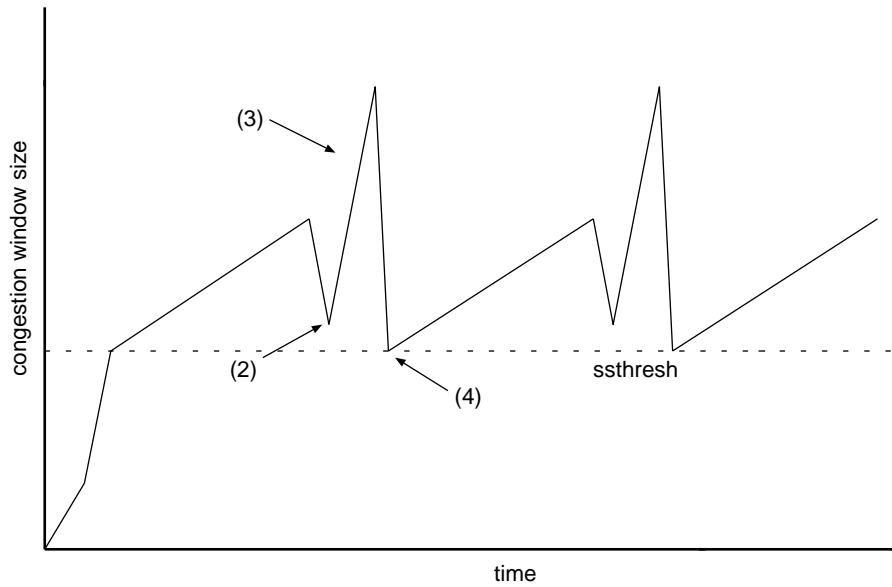


図 7: TCP Reno 方式の輻輳ウィンドウサイズの変化

ト量を急激に減らす必要がある。すなわち、*ssthresh* をセグメント損失を検出した時点の *cwnd* の半分に再設定し、さらに輻輳ウィンドウサイズを 1 セグメントに下げて、スロースタートフェーズに移行してセグメント再送を行なう。一方、重複 ACK により損失を検出した際は、重複 ACK が返って来ているので、ネットワーク内は軽い輻輳状態であると考えられる。したがって、現在一般的に用いられている TCP Reno 方式では、重複 ACK 受信によるセグメントの再送時、つまり早期再送によるセグメント再送時には、すぐに *cwnd* を縮小するのではなく、早期回復 (fast recovery) 方式で行なわれる。以下にそのアルゴリズムの説明を行なう。

送信側が 3 連続で重複 ACK を受け取ったとき、*ssthresh* を式 (1) で示すように設定する。

$$ssthresh = \frac{cwnd}{2} \quad (1)$$

その後、損失したセグメントを再送し、*cwnd* を式 (2) で示すように設定する。

$$cwnd = ssthresh + 3 \times SMSS \quad (2)$$

ここで、SMSS はセグメントの大きさである。4 つ目以降の重複 ACK を受け取ったときは、

$cwnd$ を式 (3) によって大きくしていく。

$$cwnd = cwnd + SMSS \quad (3)$$

以上の結果、 $cwnd$ が新しいセグメントを送ることができるまで大きくなった場合は、そのセグメントを送信する。新しいセグメントに対する ACK を受け取った場合は、 $cwnd$ を式 (4) のように設定する。このとき、 $cwnd$ は $ssthresh$ より大きな値であるため、これ以降は輻輳回避モードでセグメントの送信を行なう。

$$cwnd = ssthresh \quad (4)$$

図 7 に TCP Reno 方式を用いたときの輻輳ウィンドウサイズの変化の様子を示す。

2.2 無線ネットワーク環境での TCP の問題

TCP を無線ネットワーク環境で用いる場合、有線ネットワークとは異なりさまざまな問題が発生し、TCP の著しい性能劣化を引き起こすこととなる。

2.2.1 無線ネットワーク環境

無線回線は、雑音や干渉やなどの影響を受けやすいことから、有線回線よりも回線品質が悪く、TCP セグメントや ACK の損失が発生する。送信側は重複 ACK を受信し、セグメントの再送 (早期再送) を行なうが、その際には、先程説明したアルゴリズム (早期回復) にしたがって送信側の輻輳ウィンドウサイズを縮小する。また、タイムアウト時間内に ACK が送信側に到着しなければ、スロースタートフェーズに移行してセグメントの再送を行なうので、輻輳ウィンドウサイズを 1 セグメントまで縮小させることになる。回線上ではセグメントの損失は、ネットワークの輻輳状態とは無関係に発生するため、このような輻輳制御を行なう必要はない。しかしながら TCP はセグメントの損失の原因が輻輳によるものか伝送誤りによるものかを特定することができないために、一律に輻輳制御を行ってしまう。したがって、無線回線を含むようなネットワーク環境では、平均的に送信側の輻輳ウィンドウサイズが小さくなり、TCP の性能劣化を引き起こすこととなる。

また、端末が移動するような無線ネットワーク環境下では、モバイル端末が移動し通信する基地局を切替える際に、一時的なコネクションの切断が発生する。コネクションの切断は、建物などの物理的な障害によりモバイル端末に電波が届かない場合にも発生する。このようなコネクションの切断によりセグメントや ACK の損失が発生する。コネクションの切断が長期間発生すると、送信側でタイムアウト処理が行なわれ、TCP の性能劣化を引き起こすこととなる。

2.2.2 無線環境下の TCP に関する従来の研究

これまでも、無線ネットワーク環境下での TCP の性能劣化を改善するための研究が行なわれ、いくつかの方式が提案がなされている [2-4, 7-9]。以下では、無線回線上の伝送誤りによる TCP の性能劣化を改善する方式を取り上げ、その概要および問題点を説明する。

I-TCP (Indirect TCP) [2] は、送信側である固定端末と受信側であるモバイル端末間の TCP のコネクションを中間ノードである基地局で分離する方式である。固定端末からモバイル端末へ送信されたセグメントは、基地局で受信され、基地局からモバイル端末へ送信される。基地局は受信したセグメントに対して固定端末へ ACK を送信する。無線回線上の伝送誤りによるセグメント損失に対しては、基地局とモバイル端末間でコネクションが設立されているので基地局からデータをすばやく送信することができる。したがって、固定端末は無線回線上の伝送誤りによるセグメントの損失を認識する必要がないので、従来の TCP を用いるより性能劣化を改善することができる。しかし、無線回線の伝送誤りなどによりセグメントが損失しているにもかかわらず、基地局が固定端末に ACK を送信してしまい、固定端末がモバイル端末にセグメントが正しく送信されたものと誤認識してしまうことが考えられる。このことは TCP をプロトコルとして用いている多くのアプリケーションにとって問題となる。

Snoop [3] では、snoop モジュールを基地局に実装する。snoop モジュールは基地局を通過する TCP セグメントと ACK の TCP ヘッダを常に監視し、TCP セグメントの複製を蓄積する。ヘッダから得たシーケンス番号を用いて、snoop モジュールは重複 ACK から無線回線上の伝送誤りで損失したセグメントを特定し、蓄積したセグメントをモバイル端末へ再

送する。また、snoop モジュールは再送タイマーを実装しており、その時間内にモバイル端末から ACK を受信しなければセグメントを再送する。したがって、無線回線上の伝送誤りによるセグメント損失に対しては、基地局からデータをすばやく送信することができる。また、ACK が基地局に到着しないときも snoop モジュールのタイムアウト処理によりセグメントを再送することができる。これにより、TCP の性能劣化を改善することができる。

また、送信側がモバイル端末である場合に無線ネットワーク環境での伝送誤りなどによる TCP の性能劣化を改善する方式として、ELN (Explicit Loss Notification) 方式 [4] が提案されている。これは、無線回線上での損失であることを示す ELN ビットを TCP ヘッダの予約フィールドに設け、基地局において snoop モジュールが無線回線上での損失を特定し、ELN ビットを 1 にする。送信側であるモバイル端末がヘッダに設けた ELN ビットが 1 の重複 ACK を受信した時は、セグメントを再送するが輻輳ウィンドウサイズは変化させない方式である。

しかし、このように snoop モジュールを用いる場合、セグメントの再送を行なうためなどに、基地局に一定量のバッファが必要である。多くのモバイル端末が同一の基地局と通信を行なっている場合は、バッファオーバーフローが起こることも十分考えられる。また、今後 IP プロトコルとして導入される IPv6 では、IPSEC により IP ペイロードが暗号化されると、基地局などの中間ノードでのパケットのモニタを行なえない場合が考えられる [10]。

したがって、基地局での制御が必要であるこれらの方式は実現する際に解決すべき課題が多く、現実には実装が困難であると考えられる。

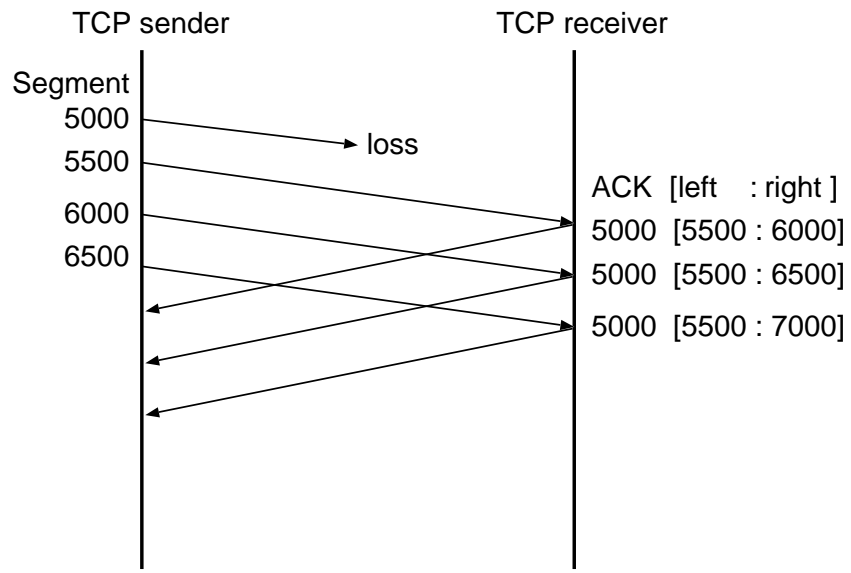
3 提案方式の概要

前述のとおり、多くのコネクションが存在する場合のバッファ管理の難しさと、データが暗号化された場合の問題を考えると、基地局などの中間ノードで TCP の制御方式を実現するのは困難であると考えられる。また、このような方式ではプロトコルの階層構造を乱すものとなっている。そこで、基地局では TCP 制御を行わず、送受信ホスト上の TCP のみの変更によって無線ネットワーク環境での TCP の性能劣化を抑える方式を提案する。提案方式は、有線ネットワーク上の送信ホストから、モバイル端末へのデータ転送を想定しており、モバイル端末が伝送誤りによるセグメントの損失を特定した場合は、その情報を送信側に伝えることによって、輻輳制御を行わないようにする。誤り発生情報の伝達は、TCP SACK 方式 [5, 6] を拡張することで実現している。本章ではまず、TCP SACK 方式を説明し、次にそれを拡張した提案方式を説明する。

3.1 TCP SACK 方式

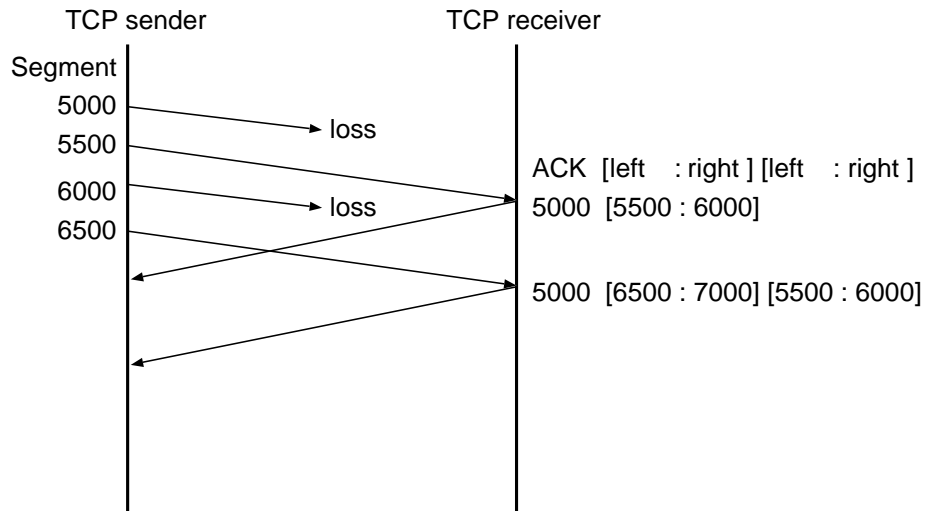
2章で説明したように、現在損失セグメントの回復方法として TCP が採用している ARQ は Go-back-N 方式である。TCP SACK 方式ではより効率の良い ARQ として Selective-repeat 方式を採用して、損失が検出されたセグメントだけを再送することによって伝送効率を上げている。TCP SACK 方式では、あらかじめ TCP コネクションを確立する際に送受信側で互いに SACK オプションを用いることを確認する。その後、受信側は ACK のオプションフィールドを用いて、明示的に到着しているセグメントのシーケンス番号を送信側に知らせる。

図 8 を用いて SACK オプションフィールドの具体例の説明をする。ここで、シーケンス番号 5000 から 6500 番のセグメントを 500 バイトの大きさで送信したものとする。図 8(a) は、最初のセグメントのみが損失した場合を表し、図 8(b) は複数のセグメントが損失した場合の SACK オプションフィールドを示している。送信ホスト側に記されている番号は、送信したセグメントのシーケンス番号を表す。また、ACK には、図 8(a) に示されるようなシーケンス番号および正常に受信したセグメントのブロックの開始と終了のシーケンス番号が示



Segment : sequence number of TCP Segment sent by TCP sender
 ACK: sequence number of ACK sent by TCP receiver
 left : first sequence number of block
 right : last sequence number of block
 block : contiguous data

(a) 最初のセグメントのみが損失した場合



(b) 複数のセグメントが損失した場合

図 8: SACK オプションフィールドの具体例

されている。また、図 8 の “loss” はセグメント損失を表しており、このセグメントは受信側には到着しないものとする。

図 8(a) では、最初のセグメントが損失しているため、その後到着したセグメントに対する ACK のシーケンス番号は 5000 番となる。5500 番のセグメントが到着した時点で受信側に到着しているのは 5500 番のセグメントだけであるため、隣接データは開始シーケンス番号が 5500 番の 500 バイトである。よって、オプションフィールドは 1 ブロックのみを使用し、開始終了シーケンスは 5500 6000 番となる。さらに、6000 番のセグメントが到着した時点で受信側に到着しているのは 5500 番、6000 番のセグメントであり、隣接データは開始シーケンスが 5500 番の 1000 バイトとなる。よって、オプションフィールドは 1 ブロックを使用し、開始終了シーケンスは 5500 6500 番となる。その後、6500 番のセグメントが到着した時点で受信側に到着しているのは 5500 番、6000 番、6500 番のセグメントであり、隣接データは開始シーケンスが 5500 番の 1500 バイトとなる。よって、オプションフィールドは 1 ブロックを使用し、開始終了シーケンスは 5500 7000 番となる。

図 8(b) でも、最初のセグメントが損失しているため、その後到着したセグメントに対する ACK のシーケンス番号は 5000 番となる。また、5500 番のセグメントが到着した時点で受信側に到着しているのは 5500 番のセグメントだけであるため、隣接データは開始シーケンス番号が 5500 番の 500 バイトである。よって、オプションフィールドは 1 ブロックを使用し、開始終了シーケンスは 5500 6000 番となる。ただし、その後図 8(b) では 6000 番のセグメントが損失している。その結果、6500 番のセグメントが到着した時点で受信側に到着しているのは 5500 番、6500 番のセグメントであり、隣接データは開始シーケンスが 5500 番の 500 バイトと開始シーケンス番号が 6500 番の 2 つである。よって、オプションフィールドは 2 ブロックを使用し、第一ブロックの開始終了シーケンスは 65007000 番となり、第二ブロックの開始終了シーケンスは 55006000 番となる。

このように、受信側に到着しているセグメントのシーケンス番号を明示的に知らせることによって、送信側は損失が発生したセグメントを知り、選択的に再送を行なうことができる。なお、TCP SACK 方式の輻輳制御は、2 章で説明した TCP Reno 方式と同様である。つまり、重複 ACK によるセグメント損失を検出した時の輻輳制御に関しては、早期再送方

式、早期回復方式で行ない、タイムアウトによるパケット損失を検出した時の輻輳制御に関しては、スロースタートフェーズで行なう。

3.2 提案方式

提案方式は、2章で説明した ELN 方式と同様に無線回線上でのセグメントの損失を送信側に通知する方式である。提案方式 (ELNSACK: Explicit Loss Notification by SACK enhancement) では SACK のオプションフィールドを用いて、この情報を送信側に通知する。これにより送信側は、無線回線上で損失したセグメントに関しては、輻輳ウィンドウサイズを下げずに損失セグメントの再送を行ない、かつ Selective-repeat 方式により損失が検出されたセグメントだけを再送することによって伝送効率を上げることが期待できる。

- 受信側 (モバイル端末) の動作

モバイル端末は無線回線の伝送単位であるフレームから誤りを検出すれば、その誤りは無線回線上の伝送誤りによって発生したものであると特定できる。セグメントの損失が発生したときは、受信側は重複 ACK を送信側に送信するが、セグメントの損失が無線回線上の伝送誤りによって発生したものであれば、重複 ACK のオプションフィールドの第一ブロックを変更した ACK を送信する。これによって送信側である固定端末は無線回線上での伝送誤りによるセグメントの損失を知ることができる。

- 送信側 (固定端末) の動作

送信側は重複 ACK によりセグメントの損失を検出する。その際に重複 ACK のシーケンス番号とオプションフィールドの第一ブロックのシーケンスを比較する。ここで、シーケンス番号が一致すればセグメントの損失が無線回線上の伝送誤りにより発生したものであると判断し、また一致しなければセグメントの損失は輻輳により発生したものである区別できる。早期再送の際に 3 つの重複 ACK について、すべての ACK のシーケンス番号とオプションフィールドの第一ブロックのシーケンス番号が一致すれば、セグメントの損失が無線回線上の伝送誤りのみで発生したものであるので、早期回復を行なわない。つまり、送信側で、しきい値 $ssthresh$ 、および輻輳ウィンドウサイズ $cwnd$ の再設定を行なわない。一方、3 つの

重複 ACK について、1 つでも ACK のシーケンス番号とオプションフィールドの第一ブロックのシーケンス番号が一致しなければ、輻輳により発生したセグメント損失があるので、早期回復を行なう。これにより、送信側で無線回線上の伝送誤りにより発生したセグメント損失に関しては輻輳ウィンドウサイズを変化させずにセグメントを再送するので、TCP 性能の改善が可能である。

図 9 を用いて提案方式の概要を説明する。ここでは、SACK は SACK オプションフィールドを使用した ACK、ELNSACK はオプションフィールドの第一ブロックを変更した ACK である。このように、セグメントを順調に受け取っている場合は、受信側は通常の ACK を送信している。ネットワーク中に輻輳が発生し、セグメントが受信側に届かない場合は、SACK を返し、再送すべきセグメントを特定する。また、この場合はウィンドウサイズを縮小して輻輳制御を行なう。無線回線上の誤りによってセグメントが損失し、それをモバイル端末が検出した場合は ELNSACK を返す。これによって送信側はウィンドウサイズを保持したままセグメントの再送を行なう。

以下に、提案方式のオプションフィールドの具体例を示す。ここでは、シーケンス番号 5000 から 6500 番のセグメントを 500 バイトの大きさで送信したものとする。図 10 に、複数のセグメントが無線回線上の伝送誤りにより損失した場合、および輻輳による損失と無線回線上の伝送誤りによる損失が混在した場合の SACK オプションフィールドを示す。図 10(a) では、すべての損失が無線回線上の伝送誤りによるものであるため、第一ブロックのシーケンス番号を ACK のシーケンス番号と同じにし、第二ブロックで受信セグメントのシーケンス番号を通知するのに使用する。なお、第二ブロック以降の使用方法は SACK 方式と同様である。

図 10(b) のように損失の原因が混在する場合においても、輻輳により損失したセグメントに対して第一ブロックから受信セグメントのシーケンス番号を通知するのに使用し、無線回線上の伝送誤りにより損失したセグメントに対して発生した重複 ACK に関しては、第一ブロックのシーケンス番号を ACK のシーケンス番号と同じにし、第二ブロック以降で受信セグメントのシーケンス番号を通知する。なお、第二ブロック以降の使用方法は SACK 方式と同様である。

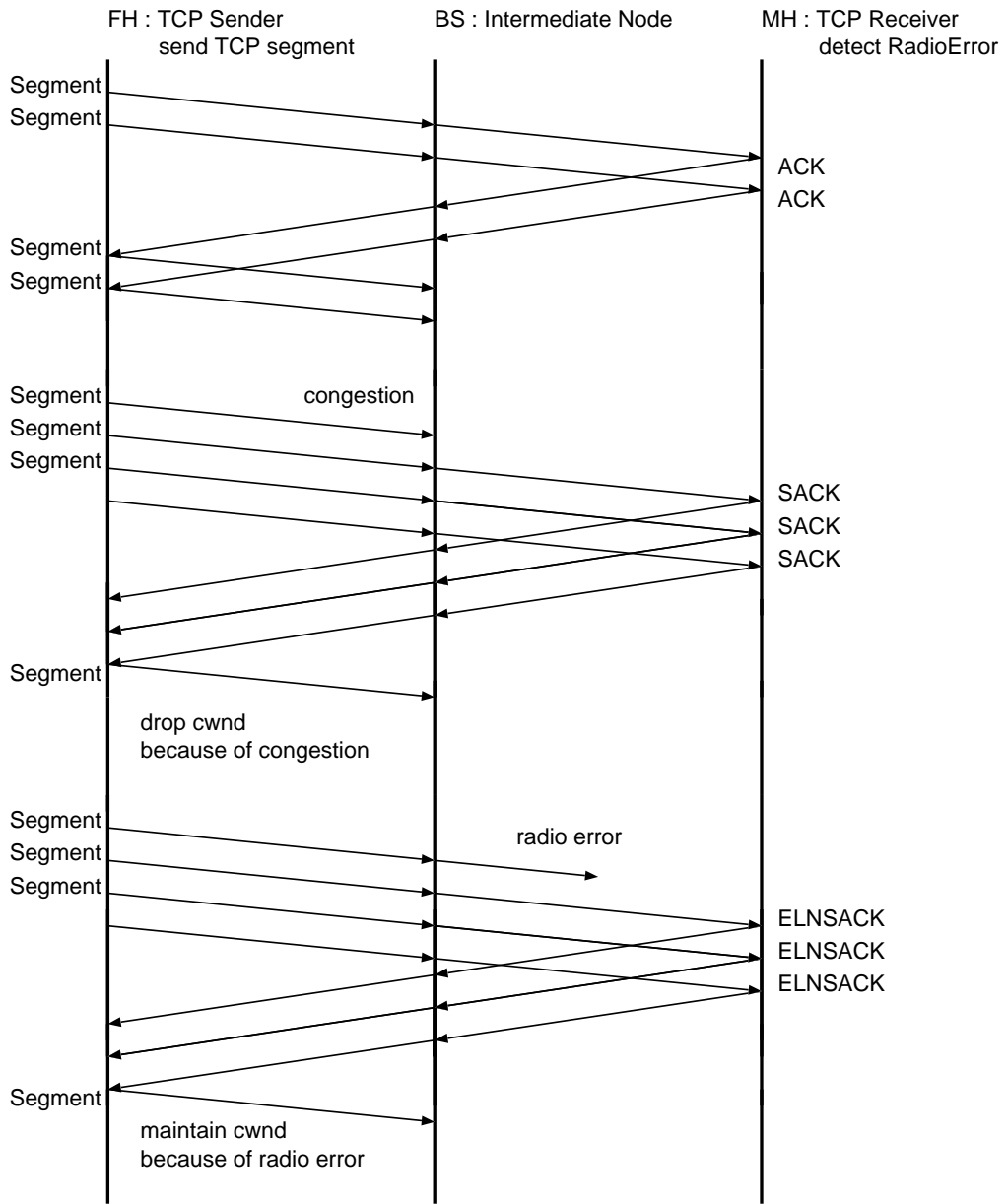
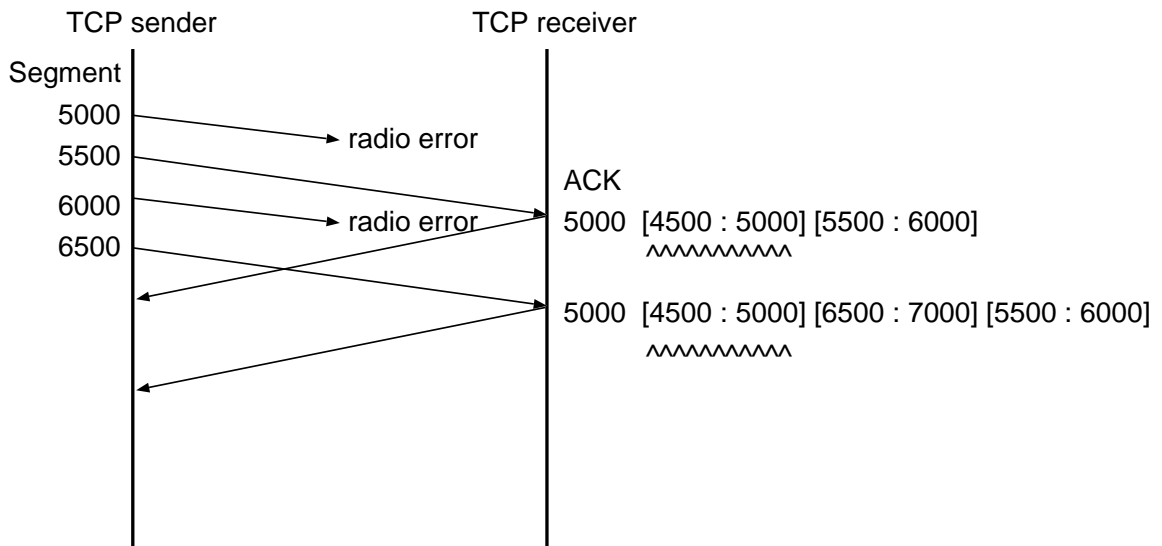
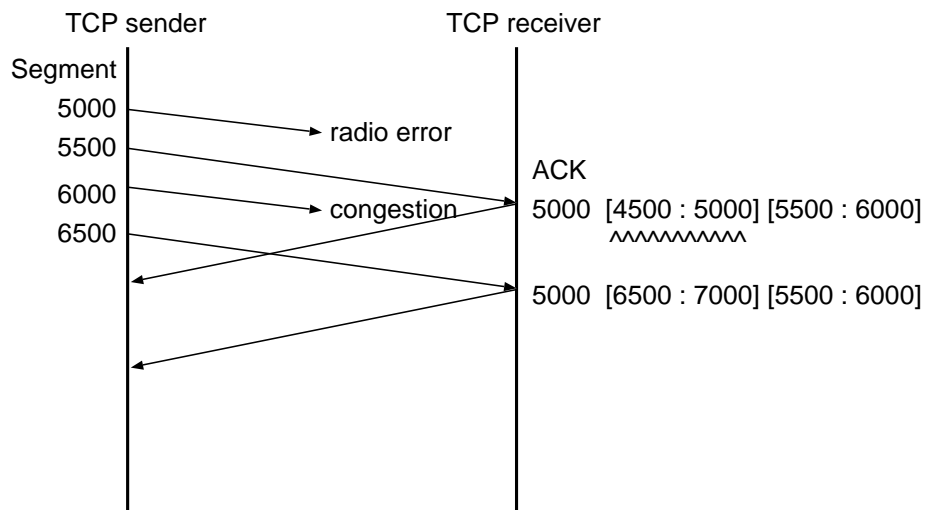


図 9: 提案方式における ACK の種類と再送方式



Segment : sequence number of TCP Segment sent by TCP sender
 ACK: sequence number of ACK sent by TCP receiver

(a) 複数のセグメントが無線回線上で損失した場合



(b) 複数のセグメントが無線回線上と輻輳で損失した場合

図 10: 提案方式オプションフィールドの具体例

このような ACK を受け取ることにより送信側は再送すべきセグメントを認識し、かつ無線回線上の伝送誤りによるセグメントの損失が発生していることを特定できる。

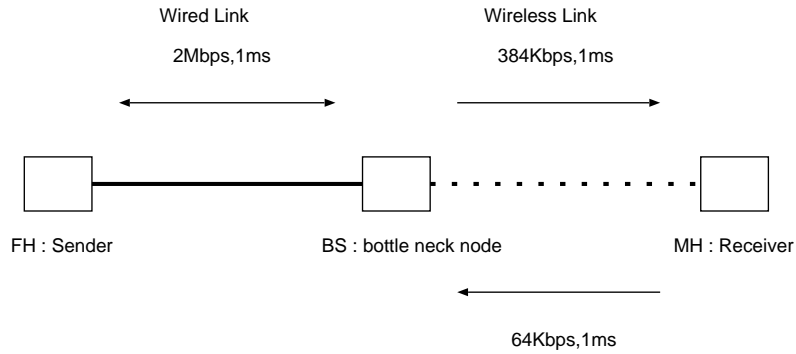


図 11: 対象とするネットワークの構成

4 シミュレーションによる評価

提案方式である ELNSACK の有効性を検証するために、シミュレーションによって現在 TCP として広く利用されている TCP Reno、および提案方式の拡張対象となった TCP SACK との性能比較を行なう。シミュレータとしては、TCP の特性評価研究でよく用いられる ns-2 [11] を採用した。

4.1 ネットワークモデル

図 11 に、シミュレーションに用いたネットワークモデルを示す。固定端末 (FH: Fixed Host) を送信側、モバイル端末 (MH: Mobile Host) を受信側とし、基地局 (BS: Base Station) を中間ノードとして、FTP の連続送信を行う。各回線の伝送誤り率および伝搬遅延時間は、図 11 に示す値を用いた [7]。無線回線の伝送速度は、IMT-2000 における歩行者を想定したものである。シミュレーションでは、最大ウィンドウサイズを 32 セグメント、基地局をボトルネックノードとしてバッファ長を 15 セグメントに設定し、バッファオーバーフローによる輻輳をシミュレートしている。また、無線回線上の伝送誤りはランダムに発生するものとしている。

4.2 提案方式の評価

以下、ELNSACK 方式の評価を TCP スループット、および輻輳ウィンドウサイズの変化の様子を調べることによって行なう。

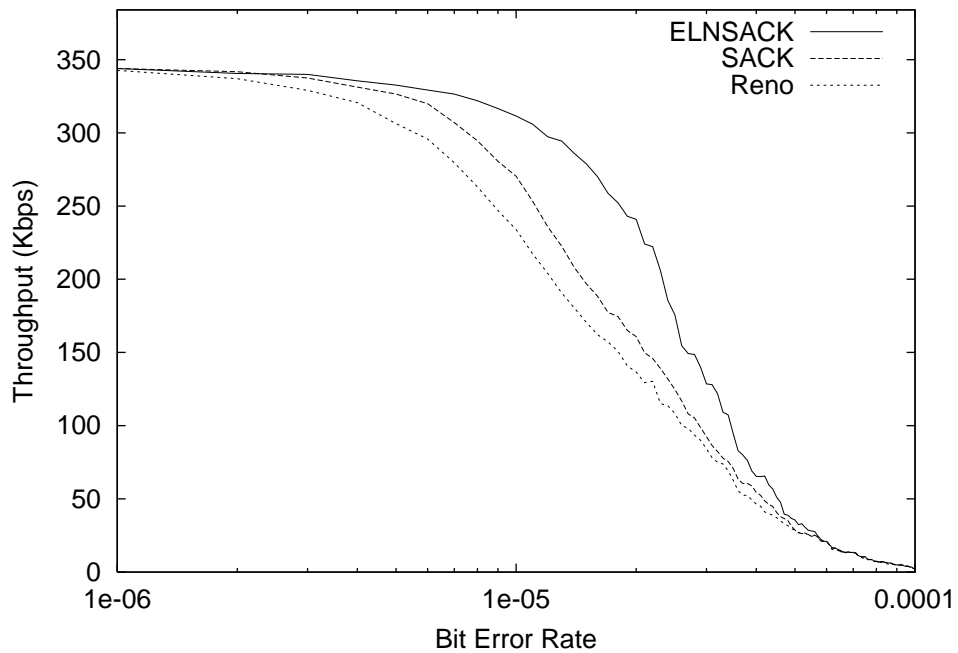
4.2.1 TCP スループット

図 12 にセグメントサイズを 512 bytes、および 1 Kbytes とし、無線回線上のビット誤り率を変化させた場合の TCP スループットのシミュレーション結果を示す。いずれのセグメントサイズにおいても、ビット誤り率が低い範囲では、ELNSACK 方式と TCP Reno、SACK 方式どの方式においてもスループットの差はあまりないが、ビット誤り率が高くなるにつれて、TCP Reno、SACK 方式のスループットの劣化が激しい。また、さらにビット誤り率が高くなると、どの方式でもスループットは小さくなってしまう。

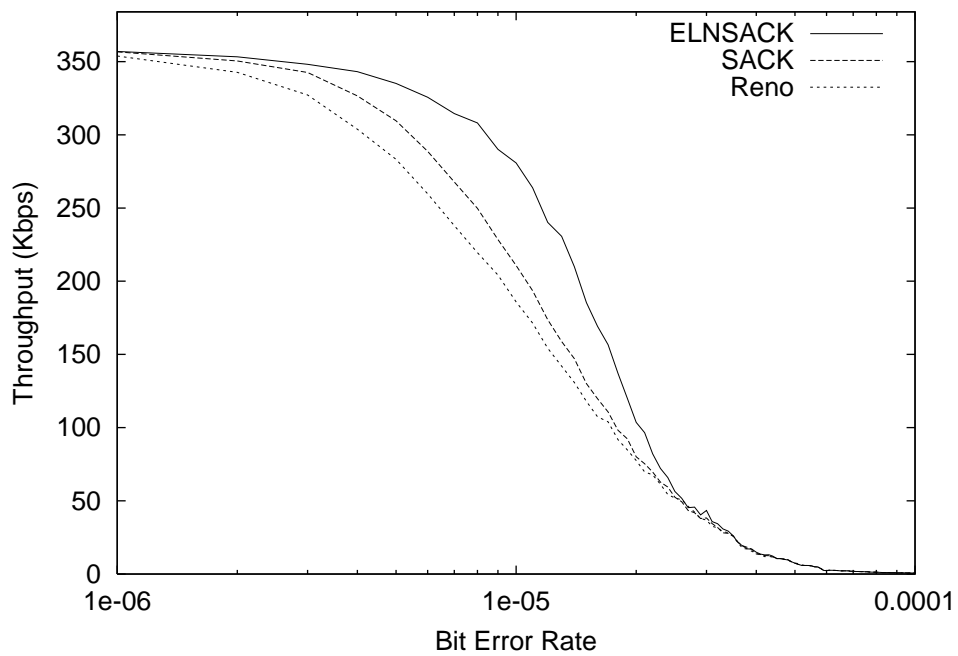
ビット誤り率が低い範囲では、無線回線上での伝送誤りにおけるセグメント損失があまり発生しないので、ネットワークにおけるセグメント損失の主な原因は輻輳によるものとなる。輻輳によるセグメント損失に関しては、どの方式においても早期再送の際に輻輳制御を行うので、スループットの差があまりないことが現れている。

一方、ビット誤り率が高くなるにつれて、無線回線上での伝送誤りによるセグメント損失が発生しやすくなり、それがセグメント損失の主な原因となってくる。ELNSACK 方式では、無線回線上のセグメント損失に関しては輻輳制御を行わないので、ネットワークに送出するデータ量を減らさない。しかし、TCP Reno や SACK 方式では無線回線上の損失に関しても輻輳制御を行うため、ネットワークに送出するデータ量を不必要に減らしている。この差がスループットの差となっている。

さらにビット誤り率が高い場合、無線回線上でのセグメント損失が頻繁に発生し、受信側でのセグメントの受信が行われにくくなる。セグメントが到着しないと受信側から ACK を送信することができなくなり、送信側に ACK が到着しなくなる。すなわち、タイムアウト時間内に ACK が到着しなくなるため、送信側では無線回線上でのセグメント損失をタイムアウトにより検出することになる。この場合のセグメント再送に関しては、輻輳ウィンドウ



(a) セグメントサイズ: 512 bytes



(b) セグメントサイズ: 1 Kbytes

図 12: TCP スループット

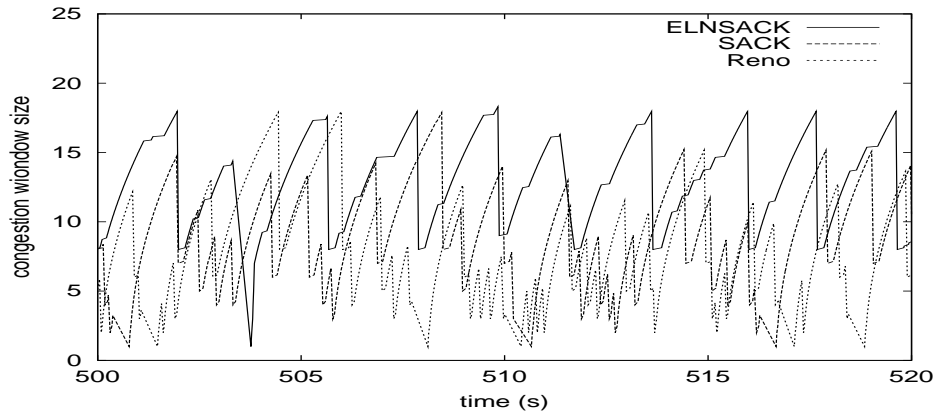
サイズを1セグメントに下げてもスロースタートフェーズでセグメントの再送を行う。タイムアウト処理に関しては、いずれの方式においてもセグメント損失の原因にかかわらず輻輳制御を行うので、スループットの劣化が非常に激しくなっていると同時に、方式間の差が表れていないことがわかる。

4.2.2 輻輳ウィンドウサイズの変化

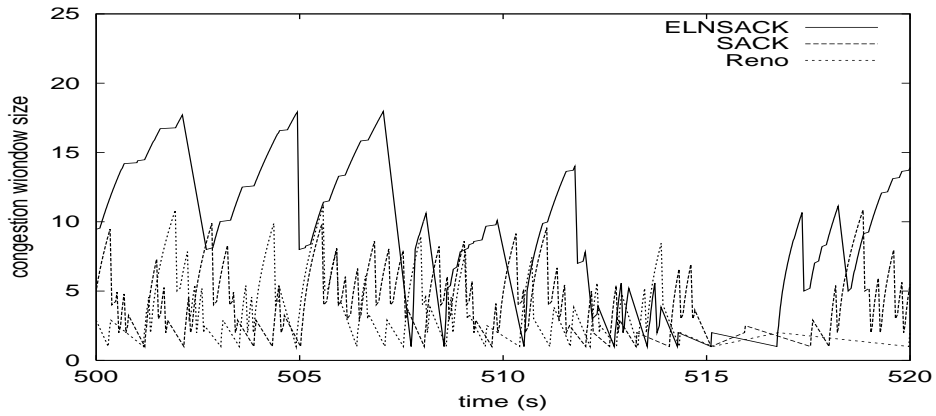
次に図13に、セグメントサイズが512 bytesのときの、各ビット誤り率における輻輳ウィンドウサイズの変化を示す。図13(a)は、ビット誤り率が 5×10^{-6} の場合における各方式の輻輳ウィンドウサイズの変化の様子を示している。無線回線上でのセグメント損失があまり発生していないので、ELNSACK方式とTCP Reno、SACK方式での輻輳ウィンドウサイズの大きさにそれほど差はなく、ネットワーク送られるデータ量にも大きな差がないことが確認できる。しかし、ELNSACK方式に比べ、TCP RenoとSACK方式では、輻輳ウィンドウサイズの変化の様子が大きく異なることが確認できる。TCP RenoやSACK方式では、損失の原因によらず輻輳制御を行うので、輻輳ウィンドウサイズの減少する機会が多い。一方、ELNSACK方式では、無線回線上のセグメント損失に関しては輻輳制御を行わず、輻輳によるセグメント損失に関してのみ輻輳制御を行うためである。

図13(b)はビット誤り率が 15×10^{-6} の場合における各方式での輻輳ウィンドウサイズの変化の様子を示している。この場合は、無線回線上でのセグメント損失がセグメント損失の原因となる。したがって、ELNSACK方式では、早期再送の際に輻輳ウィンドウサイズを減少させない。一方、TCP RenoやSACK方式では、早期再送の際に輻輳ウィンドウサイズを減少させるので、ELNSACK方式に比べて全体的に輻輳ウィンドウサイズが小さく、その結果、スループットに差が生じていることが確認できる。

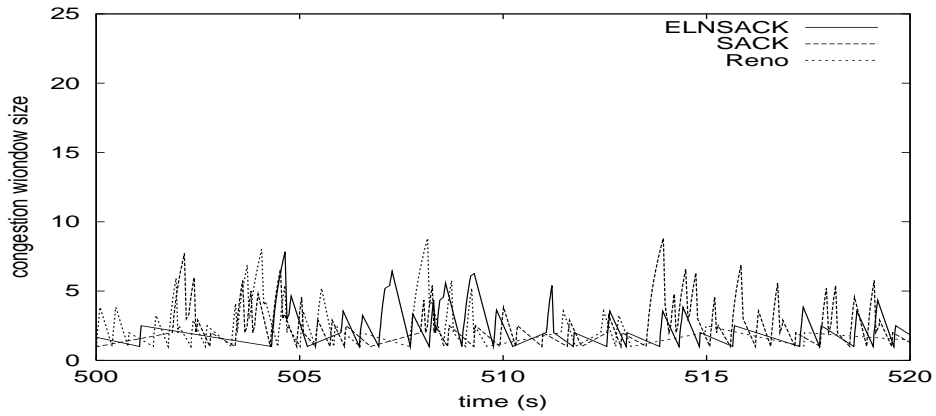
図13(c)はビット誤り率が 30×10^{-6} の場合における各方式での輻輳ウィンドウサイズの変化の様子を示している。この場合は、無線回線上でのセグメント損失が頻繁に発生しており、タイムアウトによるセグメント損失の検出が行われ、輻輳ウィンドウサイズが1セグメントに減少する。ELNSACK方式とTCP Reno、SACK方式どの方式でもタイムアウトによるセグメント損失の検出が頻繁に発生するため、輻輳ウィンドウサイズが頻繁に1セグメ



(a) ビット誤り率: 5×10^{-6}



(b) ビット誤り率: 15×10^{-6}



(c) ビット誤り率: 30×10^{-6}

図 13: 輻輳ウィンドウサイズの変化 (セグメントサイズ: 512 bytes)

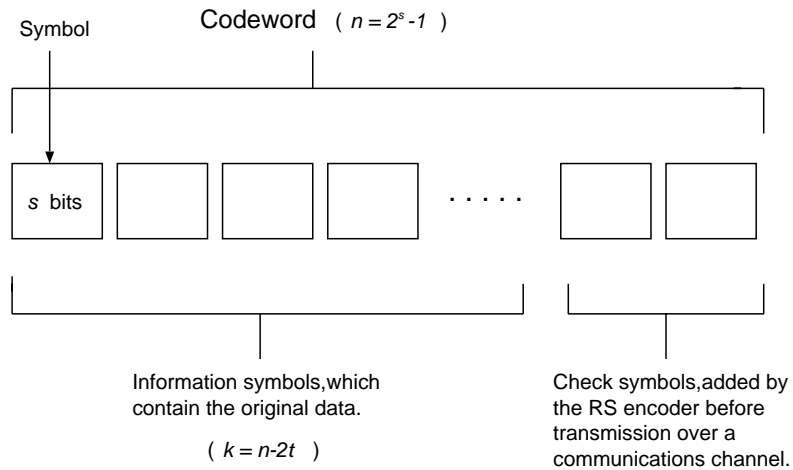


図 14: リードソロモン符号のブロック構成

ントに縮小され、その結果、ネットワークに送出されるデータ量も非常に少なくなっていることが確認できる。

4.3 FEC の適用

先に示したように、無線回線上での伝送誤り率が高くなると、セグメント損失が頻繁に発生し、タイムアウトによってウィンドウサイズが縮小されるため、性能劣化が著しくなる。そのような場合には、一般的に FEC (Forward Error Correction) を適用して、受信側で無線回線上での伝送誤りを訂正して、性能劣化を改善する手法が用いられる [12]。ここでは、誤り訂正符号としてブロック符号であるリードソロモン (RS: Reed Solomon) 符号を適用することを考える。RS 符号は、シンボル単位での誤り訂正を行なう符号であり、RS 符号では 1 シンボルを s ビットで構成して符号化する。RS(n, k) は符号長 $n = 2^s - 1$ であり、 t 個までのシンボルが訂正可能である。また、情報シンボル $k = n - 2t$ となる (図 14)。

4.3.1 セグメント誤り率の算出

IMT-2000 では、変調方式として QPSK (Quaternary Phase Shift Keying) 方式の採用が考えられている。QPSK 変調方式を用いた時のビット誤り率 p_b は、 E_b を 1 ビットのエネ

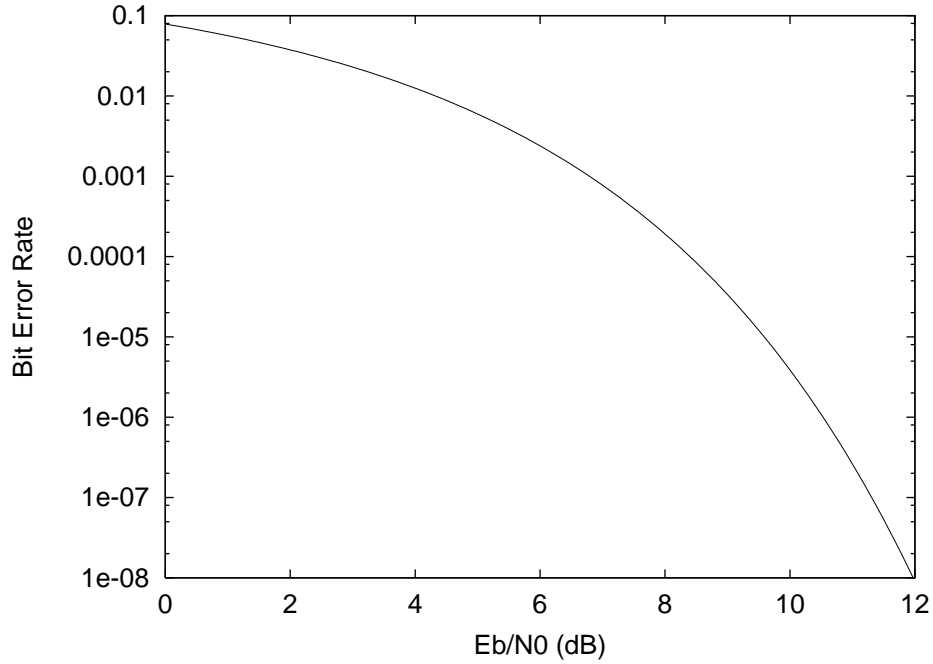


図 15: QPSK 変調でのビット誤り率特性

ルギー、 N_0 を雑音電力密度として式 (5) のようになる [13]。

$$p_b = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \quad (5)$$

ここで、QPSK 変調方式を用いたときのビット誤り率を図 15 に示す。セグメントサイズを N_T 、フレームサイズを N_L とすると、1 セグメントは $f = \lceil N_T/N_L \rceil$ 個のフレームに分割されて無線回線上で伝送される¹⁶。誤り訂正符号による訂正可能数を c 、1 フレームのビット長を l とし、フレーム誤り率 p_f は式 (6) のようになる。

$$p_f = 1 - \sum_{i=0}^c \binom{l}{i} p_b^i (1 - p_b)^{l-i} \quad (6)$$

よって、セグメント棄却率 p_s は式 (7) のようになる。

$$p_s = 1 - (1 - p_f)^f \quad (7)$$

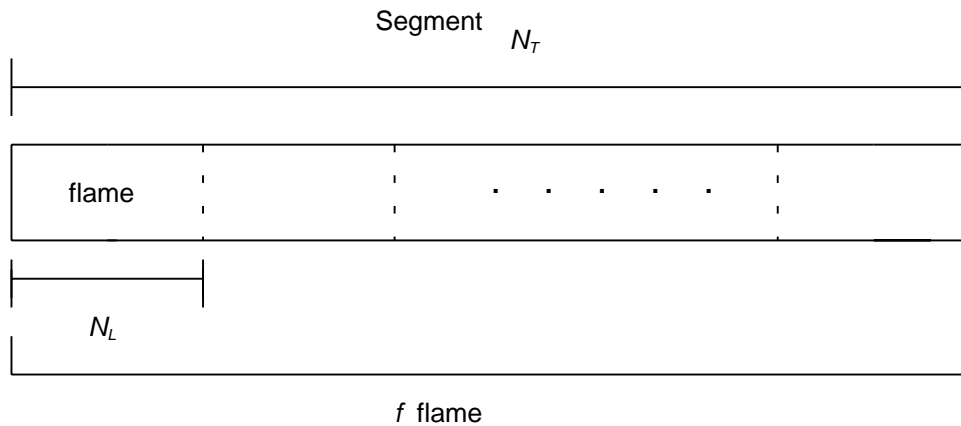
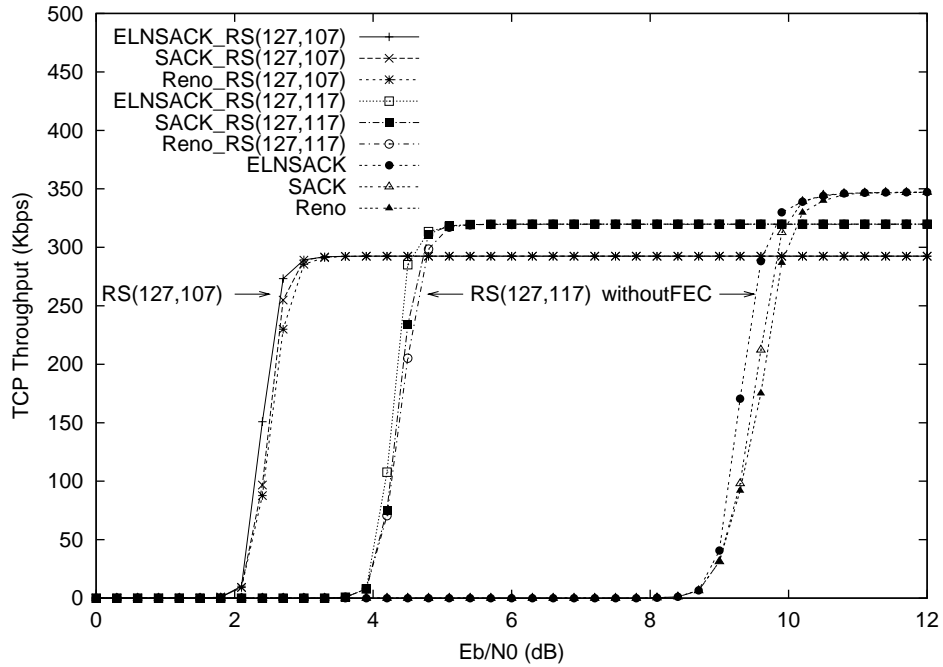


図 16: セグメントとフレームの関係

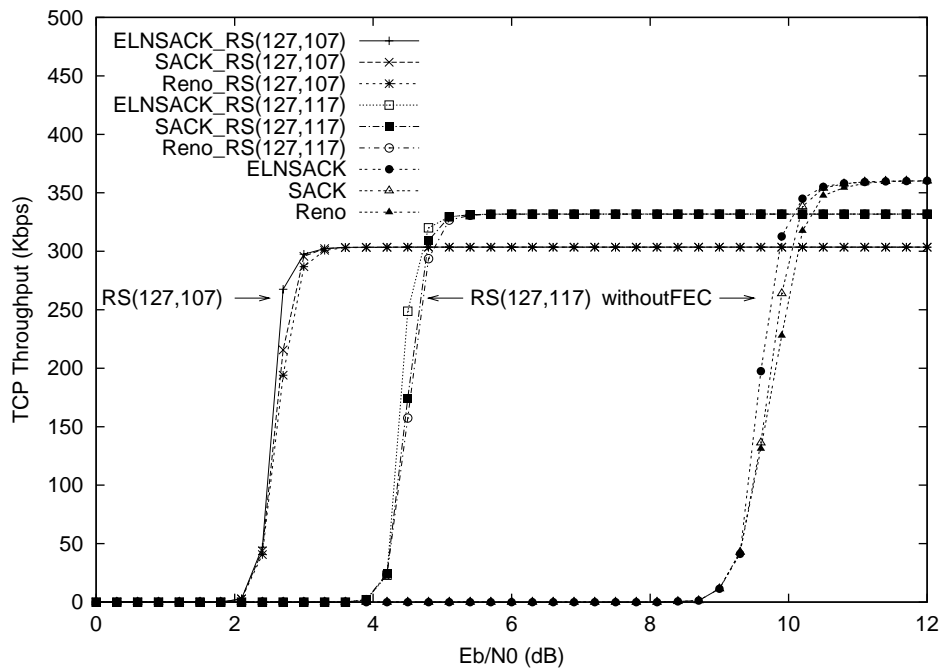
4.3.2 シミュレーションによる評価

ここでは、RS 符号として RS(127,117) および RS(127,107) を適用する。すなわち、1 フレームを構成する符号長を 127 ビットとし、情報シンボルをそれぞれは 117 ビット、107 ビットとする。これらの場合、誤り訂正能力は、それぞれ 5 ビットおよび 10 ビットである。なお、ここでは ACK は損失しないものと仮定している。

図 17 にセグメントサイズを 512 bytes、1 Kbytes としたときの、各無線回線上のノイズレベル E_b/N_0 における TCP スループットのシミュレーション結果を示す。図 17 において、スループットが劣化し始める点を劣化開始点と呼ぶことにする。FEC を適用することで、劣化開始点は RS(127,117) で約 6.0 dB、RS(127,107) で約 7.5 dB 改善されたことが確認できる。FEC の訂正能力を上げることによって、無線回線上の伝送誤り率が非常に高い場合でも TCP の性能劣化開始点を改善できる。しかし、FEC の冗長ビットによって帯域が占有され、伝送効率が低下していることも確認できる。また、RS 符号を適用することでスループットの劣化は E_b/N_0 の低下に対して劣化開始点から急激に劣化していることが確認できる。これは、RS 符号が訂正可能なビット誤りであれば、フレーム誤りが発生しないが、訂正可能なビット誤りを越えた時には、図 18 に示されているようにノイズレベルの変化に対してビット誤り率が高くなる。したがって、フレーム誤りが増え、その結果セグメント誤りが増えるためである。セグメント誤り率が高くなると、受信側にセグメントが到着しにくく



(a) セグメントサイズ: 512 bytes



(b) セグメントサイズ: 1 Kbytes

図 17: RS 符号を適用したときの TCP スループット

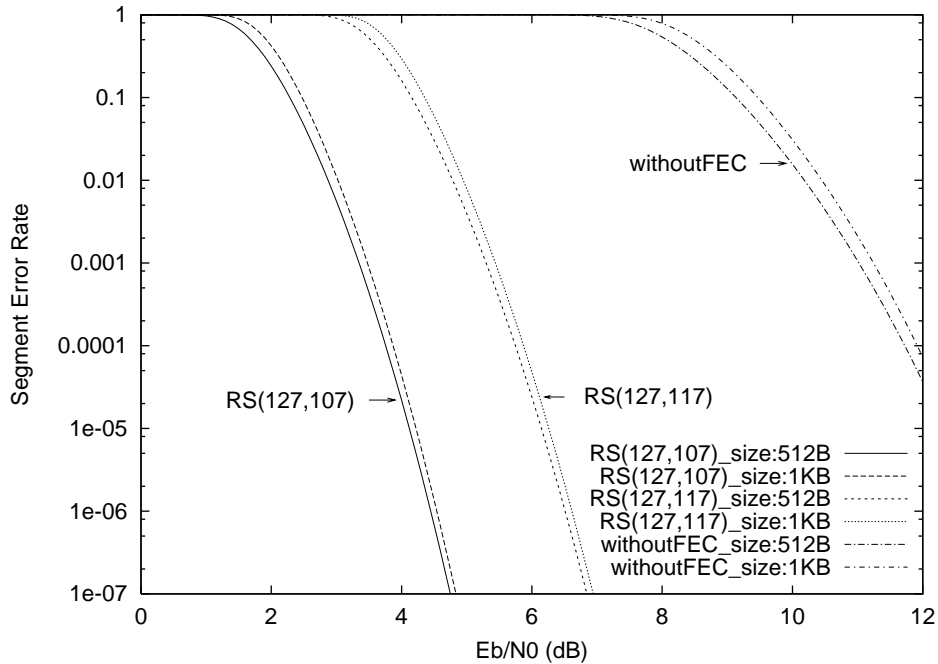
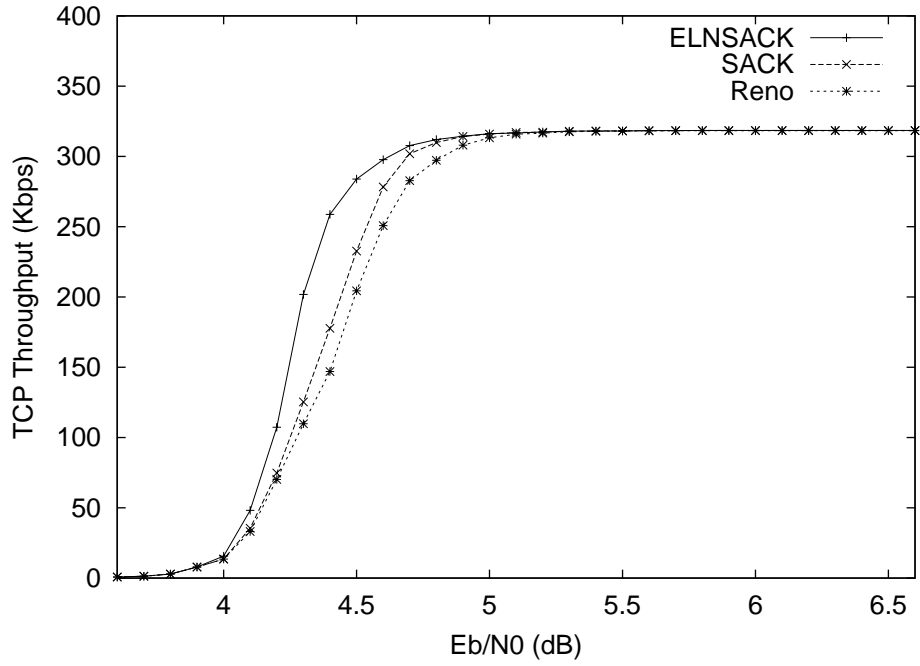


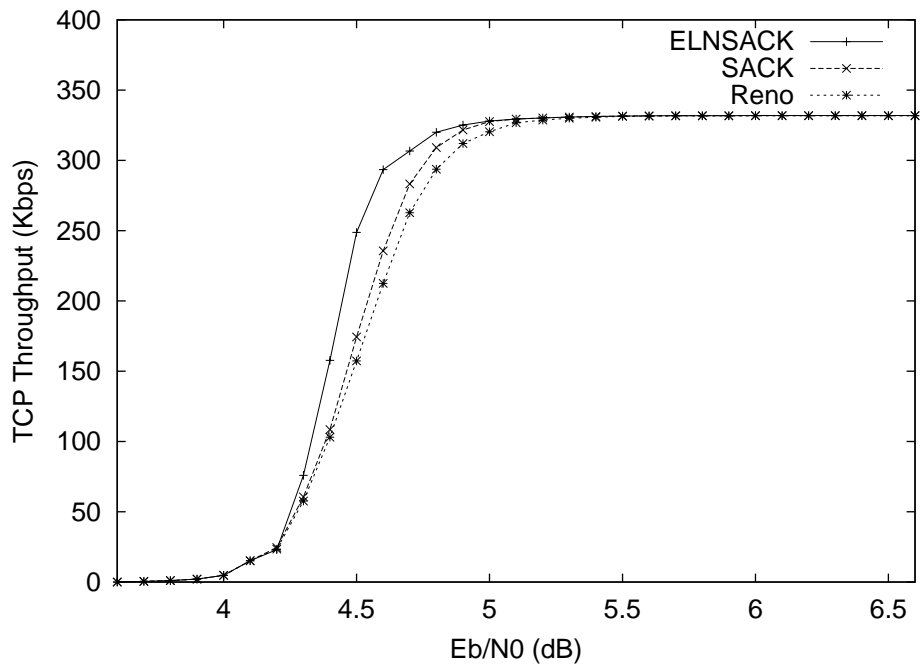
図 18: セグメント誤り率

なり、その結果、送信側にタイムアウト時間内に ACK が到着しなくなり、タイムアウト処理によるセグメントの再送が引き起こされる。したがって、ELNSACK 方式と TCP Reno、SACK 方式ともにスループットが急激に劣化することとなる。

しかし、図 19 や図 20 に示されているように、ELNSACK 方式を用いることによって、FEC を用いた場合の劣化開始点からの急激な劣化を改善することも確認できる。これは、ELNSACK 方式では、早期再送の際に無線回線上のセグメント損失に関しては輻輳制御を行なわないので、ネットワークの送出データ量を不必要に減少させていないためである。実際の無線ネットワーク環境を考えると、無線回線の品質の変動が大きく、FEC の訂正能力を越えるような無線回線上での伝送誤りが発生する状況が考えられる。そのような場合でも ELNSACK 方式を用いることによって TCP の性能劣化を抑えることができることが期待できる。

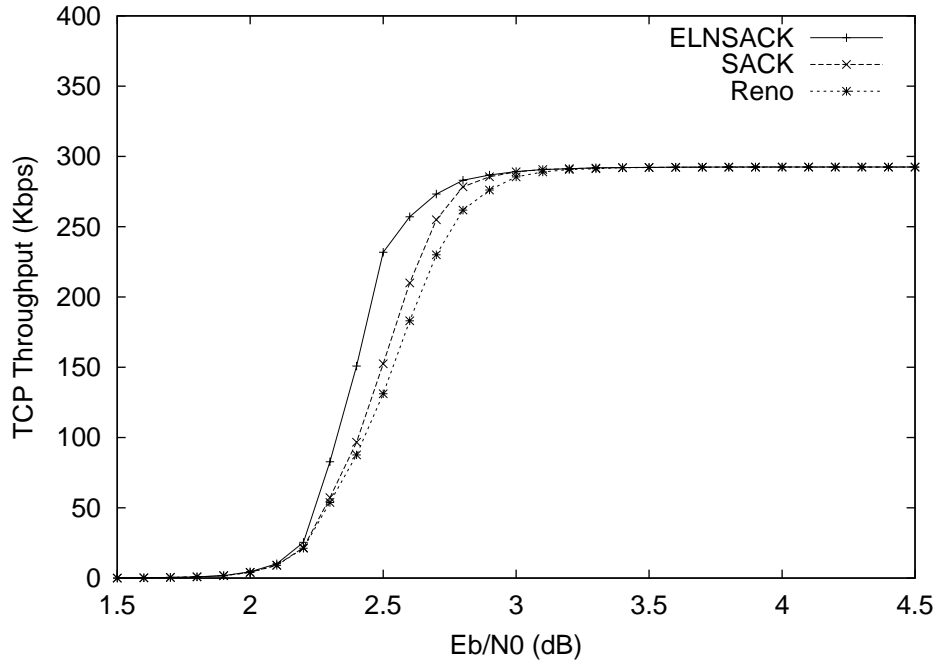


(a) セグメントサイズ: 512 bytes

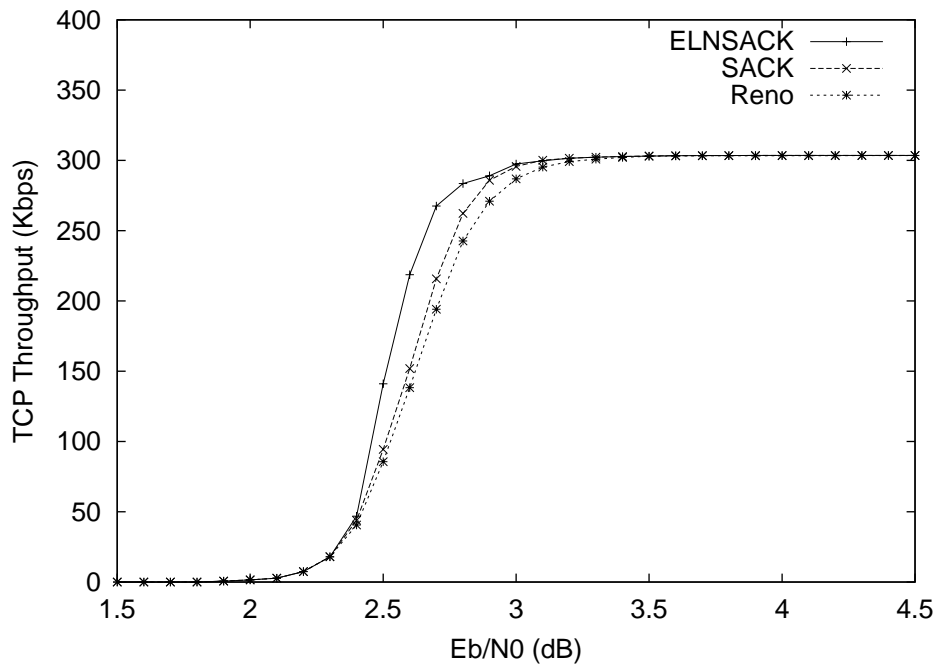


(b) セグメントサイズ: 1 Kbytes

図 19: RS(127,117) を適用したときの TCP スループット



(a) セグメントサイズ: 512 bytes



(b) セグメントサイズ: 1 Kbytes

図 20: RS(127,107) を適用したときの TCP スループット

5 おわりに

本報告では、無線ネットワーク環境において、無線回線上でのセグメント損失に起因する TCP の性能劣化を改善するための方式である ELNSACK 方式を提案した。本方式は、有線ネットワーク側のホストからモバイル端末へのデータの転送を想定している。モバイル端末において伝送誤りが検出された場合には、それを送信側に知らせることにより、セグメント損失が発生しても送信ホストにおける輻輳制御を回避することができる。無線回線上の誤り発生情報の伝達は、TCP Selective Acknowledgement (SACK) 方式のオプションを拡張することによって行なう。ELNSACK において、TCP 機構の変更は送受信ホストのみであり、基地局などの中間ノードでの制御を必要としない。本報告では、シミュレーションによって、従来の TCP と提案方式の性能比較を行なった結果、スループットの向上ができることを明らかにした。また、無線回線上の誤り発生率が非常に高い場合には、FEC を適用することで性能改善を行なう場合があるが、その場合にも本方式による再送を行なうことで、性能劣化をより改善できることを明らかにした。

今後は、基地局あたりのモバイル端末数を増やしコネクションを増加させたときや、無線回線上での伝送誤りがバースト誤りで起きるとしたときの ELNSACK 方式の評価を行なう必要がある。

謝辞

本報告を終えるにあたり、御指導、御教授を頂いた村田正幸教授に深く感謝致します。また、本報告において、遠隔地ながら、直接、あるいは電話、ファックスなどで終始御指導頂いた大阪府立看護大学医療技術短期大学の菅野正嗣助教授に深く感謝致します。ならびに、適切な助言を頂いた宮原秀夫教授、サイバーメディアセンターの馬場健一助教授、宮原研究室の若宮直紀講師、サイバーメディアセンターの大崎博之助手、村田研究室の長谷川剛助手、大阪市立大学の阿多信吾助手、経済学部の荒川伸一助手に心から感謝致します。また、株式会社デンソーの三好昌弘氏には日頃から熱心な御指導を授かり心からお礼申し上げます。最後に、御協力頂いた村田研究室および宮原研究室の皆様心からお礼申し上げます。

参考文献

- [1] W. Stevens, *TCP/IP Illustrated Volume 1*. Addison-Wesley, 1994.
- [2] A. Bakre and B. R. Badrinath, “I-TCP: Indirect TCP for mobile hosts,” in *Proceedings of the 15th International Conference on Distributed Computing Systems*, pp. 136–143, Apr. 1995.
- [3] H. Balakrishnan, S. Seshan, and R. H. Katz, “Improving reliable transport and handoff performance in cellular wireless networks,” *ACM Wireless Networks*, vol. 1, pp. 469–481, Dec. 1995.
- [4] H. Balakrishnan and R. H. Katz, “Explicit loss notification and wireless web performance,” in *Proceedings of IEEE Globecom Internet Mini-Conference*, Nov. 1998.
- [5] M. Mathis, J. Mahdavi, and S. Floyd, “TCP selective acknowledgment options,” *RFC 2018*, Oct. 1996.
- [6] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, “An extension to the selective acknowledgement (SACK) option for tcp,” *RFC 2883*, July 2000.
- [7] H. Inamura and T. Ishikawa, “A TCP profile for W-CDMA: 3G wireless packet service.” available at <http://www.ietf.org/internet-drafts/draft-inamura-docomo-00.txt>.
- [8] K. Brown and S. Singh, “M-TCP: TCP for mobile cellular networks,” *ACM Computer Communications Review (CCR)*, vol. 27, pp. 19–43, July 1997.
- [9] T. Goff, J. Moronski, and D. S. Phatak, “Freeze-TCP: A true end-to-end TCP enhancement mechanism for mobile environments,” in *Proceedings of IEEE INFOCOM '2000*, pp. 1537–1545, Mar. 2000.
- [10] 西田 佳史, “無線ネットワークにおける TCP の改善に関する考察,” 情報処理学会研究報告 2000-MBL-14, pp. 39–45, Sept. 2000.

- [11] “UCB/LBNL/VINT Network Simulator - ns (version 2).” available at <http://www-mash.cs.berkeley.edu/ns/>.
- [12] A. Chockalingam and M. Zorzi, “Wireless TCP performance with link layer FEC/ARQ,” in *Proceedings of IEEE ICC '99*, vol. 2, pp. 1212–1216, June 1999.
- [13] 藤野 忠, デジタル移動通信. 昭晃堂, Mar. 2000.