

## 誤り訂正技術を用いた TCP スループット改善手法の性能評価

佐当 百合野<sup>†</sup> 古閑 宏幸<sup>††</sup> 池永 全志<sup>†††</sup>

<sup>†</sup> 北九州市立大学大学院 国際環境工学研究科

<sup>††</sup> 北九州市立大学 国際環境工学部

<sup>†††</sup> 九州工業大学大学院 工学研究院

E-mail: <sup>†</sup>yuri@net.is.env.kitakyu-u.ac.jp, <sup>††</sup>h.koga@kitakyu-u.ac.jp, <sup>†††</sup>ike@ecs.kyutech.ac.jp

あらまし TCP を利用した通信においてパケットロスが発生した場合、特に高遅延環境下ではロスパケットの回復に時間を要するため、伝送速度を低く抑える時間が長くなり、スループット特性が劣化する恐れがある。そこで我々は誤り訂正技術 (FEC) を TCP の動作に適用することで、ロスパケット回復期間における伝送速度の低減を回避し、TCP のスループット特性を向上させるための手法の初期的な検討を行ってきている。本研究では、TCP に FEC を単純に適用した場合の課題を解決し、TCP の通信性能を改善するための手法を提案する。具体的には、冗長度や確認応答の制御手法について検討し、その有効性をシミュレーション評価により明らかにする。

キーワード 誤り訂正技術, TCP, 冗長度制御, 輻輳制御, FEC

## Performance Evaluation of TCP Throughput Improvement Methods Using Forward Error Correction

Yurino SATO<sup>†</sup>, Hiroyuki KOGA<sup>††</sup>, and Takeshi IKENAGA<sup>†††</sup>

<sup>†</sup> Graduate School of Environmental Engineering, University of Kitakyushu

<sup>††</sup> Faculty of Environmental Engineering, University of Kitakyushu

<sup>†††</sup> Graduate School of Engineering, Kyushu Institute of Technology

E-mail: <sup>†</sup>yuri@net.is.env.kitakyu-u.ac.jp, <sup>††</sup>h.koga@kitakyu-u.ac.jp, <sup>†††</sup>ike@ecs.kyutech.ac.jp

**Abstract** The growth of the Internet now allows exchange of large amounts of data through broadband networks. However, although TCP is still commonly used as a reliable data transmission protocol, packet losses significantly degrade TCP performance in a high latency environment, such as broadband networks. This is because TCP keeps the transmission rate low while lost packets are recovered. To prevent this problem, we need to use forward error correction (FEC) that enables a sender to transmit packets with redundant information to correct transmission errors at a receiver. However, FEC is restrictively used for TCP communication where transmission rates change a lot because it is typically used for UDP communication, which has a constant transmission rate. In this study, we propose a scheme to apply FEC technology to the entire TCP operation in order to improve TCP throughput. Specifically, we discuss the ways to determine redundancy level and return acknowledgments. Furthermore, we showed the effectiveness of the proposed scheme through simulation evaluations.

**Key words** Error Correction, TCP, Redundancy Control, Congestion Control, FEC

### 1. はじめに

近年では、通信技術が発展し高速通信回線を利用した動画や音声などの大容量データ転送サービスを実現するブロードバンド通信が急速に普及している。それに伴って、通信のグローバル化が急速に進行し、高遅延な環境での通信が頻繁に行われるようになった。このようなインターネットサービスのブ

ロードバンド化・グローバル化にも関わらず、インターネットでは従来より信頼性のあるデータ転送プロトコルとして TCP (Transmission Control Protocol) [1] が広く用いられている。TCP はネットワークの状況に応じて伝送速度を調整する輻輳制御によって効率の良い通信を提供する。具体的には、ネットワークの輻輳状態をパケットロスにより検出すると、伝送速度が大きすぎると判断し、伝送速度を下げ、ロスパケットの回復

を行う。その回復中は伝送速度を低く保ち、回復後に可用帯域を探るため伝送速度を上げ始める。そのため、特に高遅延環境下では、ロスパケットの回復に時間を要するため、伝送速度を低く抑える時間が長くなり、スループット特性が劣化する。

そこでロスパケット回復時間を抑制するため、あらかじめ回復用の冗長情報を送り受信側でロスパケットを回復する誤り訂正技術 (Forward Error Correction) [2] を TCP に導入することを考える。FEC はデータパケットに加え、回復に利用するための冗長情報を送信し、データパケットがロスした場合はその冗長情報を利用して回復を行うため、再送によるロスパケットの回復が不要となる。しかし、ロスパケットの回復確率は冗長情報量に依存するため、パケットロス率に合わせた冗長情報を付与する必要があり、必要以上に冗長情報を付与した場合、ネットワークに負荷をかけることになる。そのため、ネットワークの状況に応じた冗長情報量の制御が必要となる。一般的に FEC は UDP (User Datagram Protocol) を用いた CBR (Constant Bit Rate) 通信でよく利用される。CBR 通信の伝送速度は一定であるため、冗長情報量の制御が比較的容易だからである。しかし、TCP は伝送速度が大きく変動するため、最適な冗長情報量の制御が困難である。このことから、TCP に FEC を適用する手法はいくつか提案されているが、一定の伝送速度で通信を行う場合やパケットロス後ある程度まで伝送速度が回復する間のみ適用する場合など限定的な適用にとどまっている。

我々は FEC を常に TCP の動作全体に適用することで、ロスパケット回復期間における伝送速度の低減を回避し、TCP の通信性能を向上させるための手法の初期的な検討を行ってきた [3], [4]。本研究では、TCP に FEC を単純に適用した場合の課題を解決し、TCP の通信性能を改善するための手法を提案し、その基本特性を明らかにする。具体的には、冗長情報量の制御や FEC を有効利用するための ACK 制御などについて検討し、その有効性をシミュレーション評価により明らかにする。

## 2. 関連研究

本節では、FEC を用いた関連研究について説明する。

FEC はあらかじめ回復用の冗長情報を送信し、受信側でその冗長情報によってロスパケットを回復できる技術である。スループット特性向上を目的として、これまでに冗長情報量を調整する様々な FEC アルゴリズムが提案されている。FEC アルゴリズムは、2つのグループに分けることができる。1つはネットワーク状態に関わらず、固定数の冗長パケットを送信する方法である。関連研究 [5] は無線ネットワーク環境での映像伝送に着目し、ビデオフレームの種類に応じて冗長パケットを付与することでビデオサービスの品質を向上している。もう1つの FEC アルゴリズムは、ネットワークの状況に応じて、冗長パケットの付与数を動的に決定する方法である [6]–[11]。関連研究 [6], [7] はパケットが連続してロスした際に、輻輳制御に起因する伝送速度が連続的に減少する問題に焦点を当てている。関連研究 [6] では、パケットロスを検出した後ある程度まで伝送速度を回復する間のみ TCP に FEC を適用する方法を提案

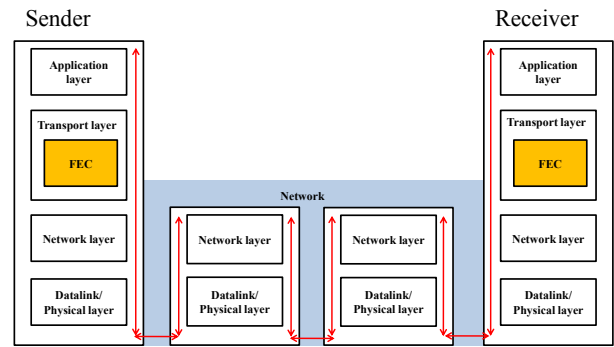


図 1 FEC 技術を利用した通信

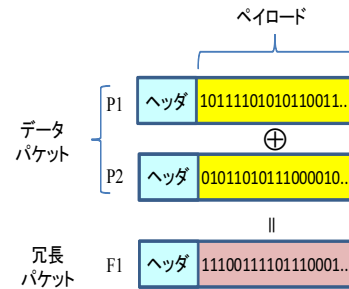


図 2 データパケットと冗長パケット

している。それによって、伝送速度の低減及び冗長情報量を抑制している。関連研究 [7] では、低遅延環境でのストリーミングの品質を維持するために ARQ メカニズムと FEC 技術を組み合わせる方法を提案している。関連研究 [8] では、輻輳制御により急激に伝送速度が低下している間に FEC を適用し、パケットロスや再送回数を抑制している。特にストリーミング配信サービスに必要な伝送速度を維持することで、高い通信性能が得られることを明らかにしている。関連研究 [9] では、高遅延環境下での短期的なバーストロスによって失われたパケットの回復に時間がかかる問題に着目している。これは、バーストロスが発生した場合に、TCP は短期または長期の輻輳状態を区別することができないためである。この問題を解決するために、関連研究 [9] では、短期輻輳を識別するために明示的輻輳通知 (ECN) を使用する TCP の輻輳制御を提案している。この方法は、ECN によってパケットロスを検出し、失われたパケットの数に比例した冗長パケットを付与することで、高遅延広帯域な環境において高いスループットを獲得することを明らかにしている。しかしこれらの関連研究では、FEC は限定的な適用にとどまっている。

本研究では、ロスパケット回復による伝送速度低減を抑制することで TCP の通信性能をあげるため、TCP の動作全体に FEC を適用する技術を確立する。しかし、TCP に単純に FEC を適用した場合、FEC が有効に働かないことがある。そのため、本研究では FEC が有効に働かない場合の課題を解決する。

## 3. 提案手法

本研究ではロスパケットの回復時間における伝送速度の低減を回避するため、FEC を常に TCP の動作全体に適用し、通信

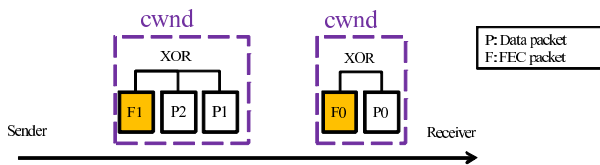


図3 冗長度の制御手法

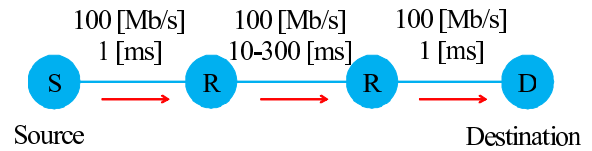


図4 シミュレーショントポロジ

性能を向上させる手法を提案する。本節では、まず提案手法の概要を説明し、その後、各制御手法について詳細な説明を行う。

### 3.1 提案手法の概要

本提案手法では、図1に示すようにエンドホストのトランスポート層にFEC機構を設ける。送信側のFEC機構において、送信データパケットから冗長パケットを生成し、受信側のFEC機構において、冗長パケットからロスパケットの復元を行う。ロスパケットの復元が目的であるため、データパケット内に冗長情報を付加するのではなく、複数のデータパケットに対する冗長パケットを生成し、フローへ挿入する。

送信側では図2に示すように各データパケットのペイロード部のビット列同士を排他的論理和の演算を行うことで冗長パケットを生成する。送信データパケットが発生する度に排他的論理和の演算を行うことで、常にその時点までのデータパケットに対する冗長パケットを生成することができる。冗長パケット生成に必要なメモリ領域も1パケット分ですみ、また単純なビット演算ですむため、処理負荷の軽減にもつながる。ここで、冗長パケットに対応するデータパケット群をFECグループと定義し、そのデータパケット数をFECグループサイズと定義する。

受信側ではあるFECグループに属するデータパケットを受信する度に、送信側と同様に排他的論理和の演算を行っておく。もし、FECグループ内でロスしたパケットが1つだけの場合は、冗長パケットを受信した際に排他的論理和の演算を行うことで、ロスパケットの復元ができる。2つ以上のデータパケットがロスした場合は復元ができない。また、ロスパケットがない場合は、冗長パケットを単に廃棄する。

以上が本提案手法の基本的な動作である。具体的な冗長度の制御手法や確認応答(ACK)の制御手法について、以降の節で説明する。

### 3.2 冗長度の制御手法

冗長度が高いほど、冗長パケットによるロスパケットの回復成功率は高くなるが、その分ネットワーク負荷も高くなってしまいうため、冗長度は適切に制御する必要がある。一般的にTCPは伝送速度が小さいときの方がパケットロスによる影響が大きい。そこで本提案手法では、伝送速度が小さい場合には冗長度を高くし、伝送速度が増加するにつれてネットワーク負荷を考慮し、冗長度を下げしていく。つまり、FECグループサイズをTCPの輻輳ウィンドウサイズ(cwnd)に合わせて制御する。具体的には、図3に示すように輻輳ウィンドウサイズから冗長パケット分を引いたcwnd-1をFECグループサイズとする。輻輳ウィンドウサイズの初期値を2とすると、データパケット1

表1 シミュレーションパラメータ

シミュレーション時間	220 [s]
バッファサイズ	50 [packet]
最大ウィンドウサイズ	65535
パケットサイズ	1500 [byte]
リンク帯域	100 [Mb/s]
試行回数	9回
ルータ間以外の遅延時間	1 [ms]
ルータ間の遅延時間	10-300 [ms]
ルータ間のパケットロス率	0.001-1 [%]
FECグループサイズの上限值	1, 10, 40

つに対して、冗長パケット1つを送信することになる。その後は、冗長パケットを送信する度に、FECグループサイズを更新していく。つまり、冗長度は $1/(cwnd-1)$ となる。

このように提案手法は伝送速度の増加とともに、冗長度を下げていくことになる。しかし、冗長度を下げすぎると、冗長パケットによるロスパケットの回復が見込めなくなってしまう。そこで、冗長度の下限値、すなわち、FECグループサイズの上限值“fmax”を導入することも考える。その場合の冗長度は $1/\min(cwnd-1, fmax)$ で表される。このFECグループサイズの上限值が性能へ与える影響についてはシミュレーション評価によって明らかにする。

### 3.3 確認応答の制御手法

前節で述べたように、本提案手法はFECグループの最後に挿入された冗長パケットによってロスパケットの回復を行う。パケットロスが発生した場合、通常のTCPのようにデータパケットを受信する度にACKを返すと、冗長パケットによる回復の前に、重複ACKによる再送が発生してしまうことがある。そこで、本提案手法では、受信側においてパケットロスを検出した場合、それ以降のデータパケットを受信してもACKの返信を保留する(ACK抑制)。その後、冗長パケットによってロスパケットが回復できた場合や2つ目のパケットロスを検出するなど回復ができないことが判断できた場合は、ACK抑制を解除し、保留していたACKを返信する。

## 4. シミュレーション評価

本稿では、提案手法の有効性を評価するため、シミュレータによって提案手法と従来手法であるTCP SACKとの比較評価を行う。本節では、シミュレーション環境、評価指標について述べる。シミュレータにはns-2 (Network simulator ver.2.35) [12]に提案手法を加えたものを使用する。

#### 4.1 シミュレーションモデル

本評価では、高遅延な通信環境を想定してシミュレーション評価を行う。シミュレーションモデルを図4に示す。今回、ノード  $R-R$  間のリンクの伝搬遅延時間の値を大きく設定することで高遅延な通信環境を実現している。

次に、表1に本シミュレーションパラメータを示す。図4内の全てのリンク帯域を 100 [Mb/s]、ノード  $S-R$  間とノード  $R-D$  間のリンクの伝搬遅延時間を 1 [ms] としている。高遅延な環境において各手法による通信性能への影響を調査するために、ノード  $R-R$  間のリンクの伝搬遅延時間を 10–300 [ms]、 $R-R$  間でのランダムロスの発生確率を 0.001–1 [%] と変化させる。また、FEC グループサイズの上限值による通信性能への影響を評価するために、FEC グループサイズの上限值 (fmax) を 1, 10, 40 と変化させる。今回のシミュレーションでは、従来手法である TCP SACK (sack) と冗長度を輻輳ウィンドウサイズで制御し、ACK 抑制を行う提案手法 (fec), fec の機能に加えて FEC グループサイズの上限值を設ける手法 (fec+fmax) との比較評価を行う。また、比較のために、ACK 抑制を行わない手法 (fec+no-dup) も用いる。シミュレーションは 9 回試行を行い、その平均値で性能評価を行う。

#### 4.2 評価指標

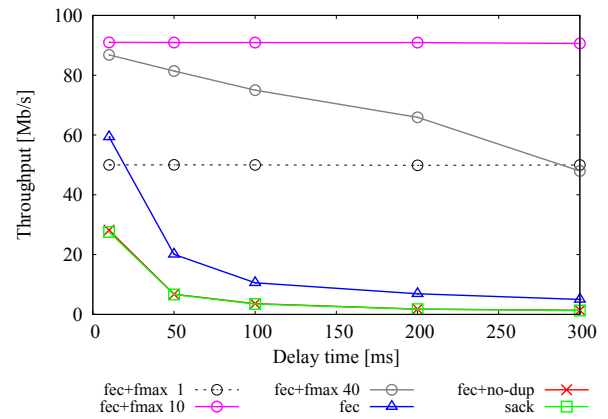
本評価では、提案手法の有効性を評価するために、評価指標としてスループットを用いる。また、そのスループット特性を分析するために、冗長率、再送率、有効回復率、タイムアウト発生回数、およびファストリカバリ発生回数についても調査する。冗長率とは全送信パケット数に対する冗長パケット数の割合を表し、再送率とは全送信パケット数に対する再送パケット数の割合を表している。有効回復率とは全冗長パケット数に対してロスパケットの回復に利用された冗長パケット数の割合を表し、例えば有効回復率が 10 [%] の場合、90 [%] の冗長パケットが無駄になったこと (パケットロスが発生しなかったか、あるいは複数パケットがロスしたため回復できなかったこと) を意味する。また、タイムアウト発生数およびファストリカバリ発生回数は、ロスパケットの回復ができず、それぞれのパケットロス検出により、データパケットの再送が発生した回数である。以上の評価指標は、試行回数で平均化した値を用いる。

### 5. シミュレーション結果

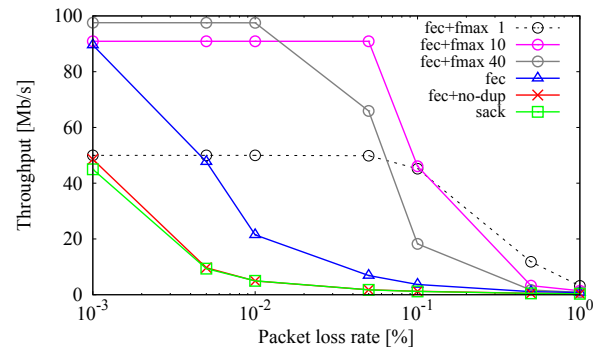
本節では、従来手法と提案手法のシミュレーション結果を比較し、提案手法の有効性について議論する。そこで、遅延時間、ランダムロス率と FEC グループサイズの上限による影響について示す。

#### 5.1 スループット特性

まず、遅延時間やパケットロス率が各提案手法や従来手法のスループット特性に与える影響について調査する。パケットロス率を 0.05 [%] とし、遅延時間を 10–300 [ms] と変化させた場合、および、遅延時間を 200 [ms] とし、パケットロス率を 0.001–1 [%] と変化させた場合の各手法のスループット特性をそれぞれ図 5(a), 5(b) に示す。“fec” は ACK 抑制は行いが FEC グループサイズに上限値を設けない提案手法で、“fec+fmax  $n$ ”



(a) 遅延時間による影響



(b) パケットロス率による影響

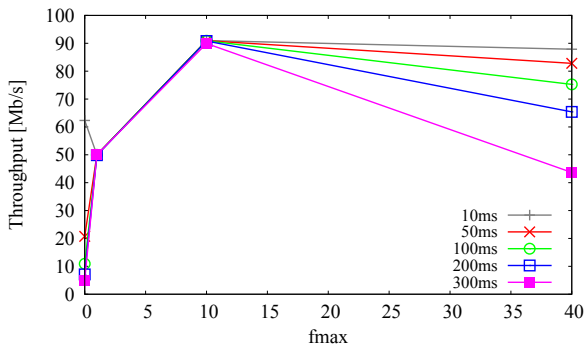
図5 スループット特性

は FEC グループサイズの上限值を  $n$  とした手法である。また、“fec+no-dup” は ACK 抑制を行わない手法で、“sack” は従来手法である。

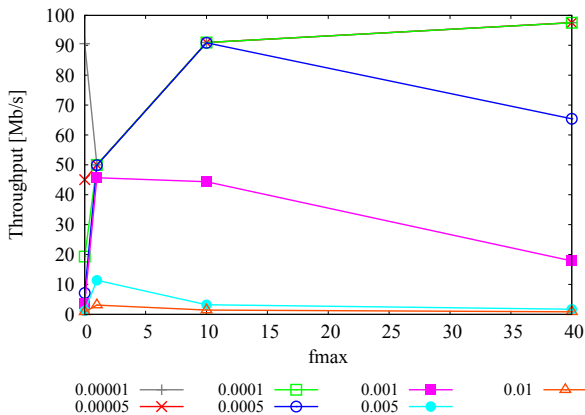
図 5(a) より、各提案手法は従来手法 sack よりも高いスループットが得られている。ただし、ACK 抑制を行わない fec+no-dup は sack と同程度のスループットしか得られていない。つまり、FEC を有効に機能させるためには、ACK を適切に制御する必要があることがわかる。ACK 抑制を行う fec は sack と比較してスループット特性が改善しているが、その改善度合いはあまり大きくない。一方、FEC グループサイズの上限值を設ける fec+fmax はスループット特性を大きく改善できることがわかる。特に、fmax を 10 と適切に設定することで、遅延時間に依存せず、高いスループットを達成している。詳細な分析については、次節で行う。

次に、図 5(b) より、提案手法はパケットロス率に関わらず、従来手法よりも高いスループットを獲得できることがわかる。各提案手法のスループット特性は図 5(a) と同様の傾向があるが、パケットロス率が小さい場合は fmax の値が 40 と大きい方が高いスループットが得られ、一方、パケットロス率が大きい場合には fmax の値が 10, 1 と小さい方が高いスループットが得られている。

そこで、fmax の値がスループット特性へ与える影響を調査するため、fmax の値を 0–40 と変化させた場合のスループット特性を図 6 に示す。ここで、fmax の値が 0 とは、上限値を設



(a) 遅延時間による影響



(b) パケットロス率による影響  
図 6 スループット特性:  $f_{max}$

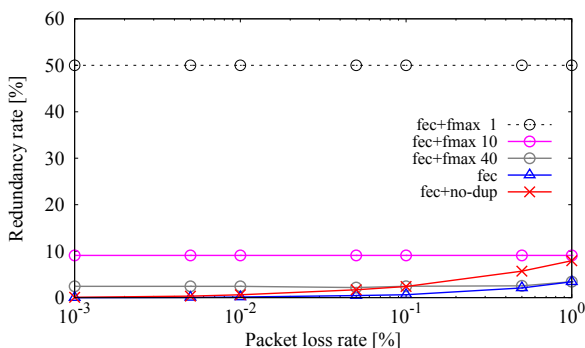


図 7 冗長率

けない場合である。図 6(a) はパケットロス率を 0.05 [%] としたときの遅延時間による影響を示しており、図 6(b) は遅延時間を 200 [ms] としたときのパケットロス率による影響を示している。これらの図より、 $f_{max}$  の値はパケットロス率に大きく影響を受け、遅延時間から受ける影響が小さいことがわかる。大きな  $f_{max}$  の値は冗長度が低くなりがちであるため、特に遅延時間が大きい場合やパケットロス率が高い場合にスループット特性が大きく低下する。また、 $f_{max}$  の値が小さすぎる場合は冗長度が高すぎるため、伝送効率が悪くなる。そのため、今回の環境では、 $f_{max}$  の値は 10 程度が適切であることがわかる。

## 5.2 性能分析

前節で、遅延時間やパケットロス率がそれぞれの手法のスループット特性に与える影響について議論した。

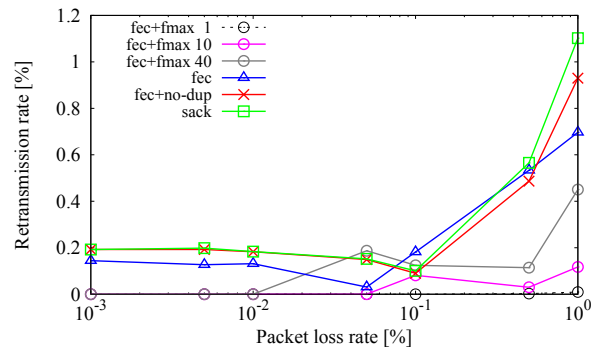


図 8 再送率

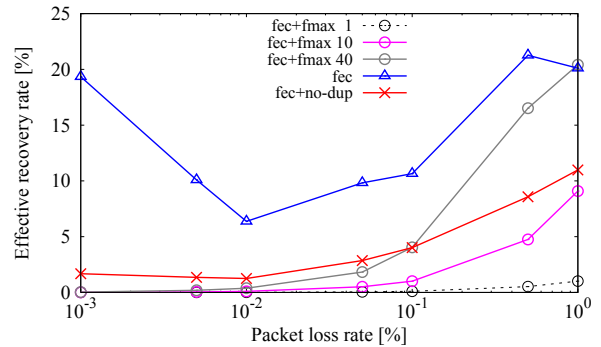


図 9 有効回復率

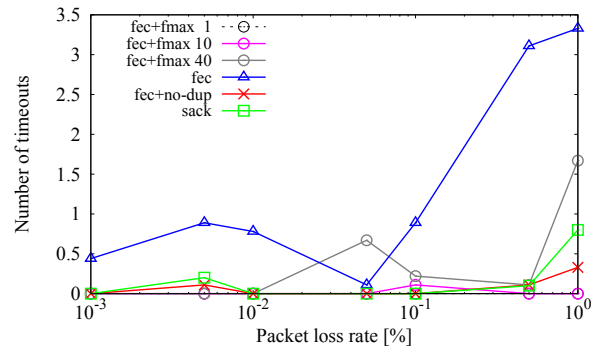


図 10 タイムアウト発生回数

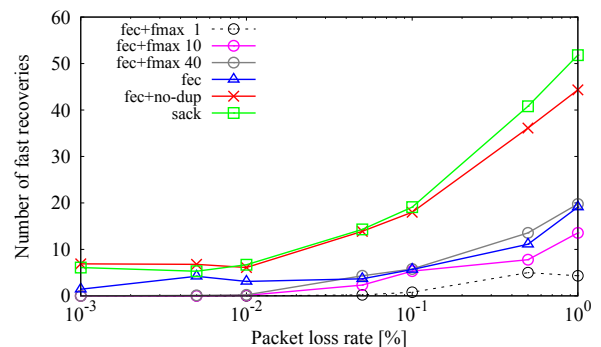


図 11 ファストリカバリ発生回数

その結果について詳細に分析するため、遅延時間を 200 [ms] とし、パケットロス率を 0.001-1 [%] と変化させた場合における冗長度、再送率、有効回復率、タイムアウト発生回数、およびファストリカバリ発生回数について調査する。

図7, 図8, 図9はそれぞれ全送信パケット数に対する冗長パケット数の割合, 全送信パケット数に対する再送パケット数の割合, および, 全冗長パケット数に対してロスパケットの回復に利用された冗長パケット数の割合を示している。また, 図10, 図11はそれぞれロスパケットの回復ができず, タイムアウトおよびファストリカバリによるパケットロス検出により, データパケットの再送が発生した回数を示している。

これらの図より, sackはファストリカバリによる再送が多く発生し, 特にパケットロス率の増加とともに非常に多くの再送が発生していることがわかる。その結果, 図5に示したように, スループット特性が大きく低下している。fec+no-dupの場合も冗長パケットによるロスパケットの回復が成功したとしても, 重複ACKによる再送(ファストリカバリ)を引き起こしてしまうため, sackと同様の再送率となっている。また, fecの場合は, ACK抑制によりファストリカバリ発生回数を小さく抑えるができてはいるが, 一方でタイムアウト発生回数が大きくなっている。これはロスパケットが冗長パケットにより回復できるかどうかを判断するためにACKを抑制する時間がFECグループサイズの増加とともに長くなってしまっているからである。その結果, 再送率を大きく低減することができず, スループット特性の改善度合いも小さくなっている。

一方で, FECグループサイズの上限値を設けるfec+fmaxの場合は, タイムアウト発生回数とファストリカバリ発生回数の両方を小さく抑え, 再送率を大きく低減することができている。その結果, スループット特性が大きく改善されている。しかし, fmaxの値が40と大きい場合, 上限値を設けないfecほどではないが, パケットロス率が高い範囲でタイムアウト発生回数が増加している。そのため, 再送率が増加し, スループットの低下を引き起こしている。また, fmaxの値が1のように小さすぎる場合は, タイムアウトやファストリカバリによる再送を劇的に抑えることができているが, 冗長率が非常に大きくなるため, 伝送効率が著しく低下する。つまり, 有効回復率も低くなり, 特にパケットロス率が低い範囲においてスループット特性の改善度合いもあまり大きくはない。したがって, fmaxの値をパケットロス率に応じて適切に設定することで, 冗長率や有効回復率を維持しつつ, 再送率を大きく低減することができ, スループット特性の改善につながる。今回の環境では, fmaxの値が10の場合がそれを達成している。

## 6. おわりに

本稿では, TCPを利用した特に高遅延な環境の通信において, ロスパケットの回復時間における伝送速度の低減を回避するため, FECを常にTCPの動作全体に適用し, 通信性能を向上させる手法を提案し, シミュレーション評価により提案手法の有効性を評価した。シミュレーション評価の結果, TCPにFECを単純に適用しただけでは改善効果は小さいが, 冗長度の制御や確認応答の制御によってスループット特性を大きく改善できることがわかった。特に, FECグループサイズの上限値を適切に設定し, ACK抑制を行うことで, 遅延時間にかかわらず, 従来手法であるTCP SACKと比較して高いスループッ

トを達成することを明らかにした。

今後は, より現実的な環境としてバーストロスが発生する環境下での評価やネットワーク状況により適切な冗長度を決定するためのアルゴリズムを検討していく予定である。

謝辞 本研究の一部は日本学術振興会における科学研究費補助金基盤研究(B)(課題番号25280028)の支援を受けている。ここに記して誠意を表す。

## 文 献

- [1] J. Postel, "Transmission Control Protocol," IETF RFC791, Sep. 1981.
- [2] M. Luby, L. Vicisano, J. Gemmell, L. Rizzo, M. Handley, and J. Crowcroft, "Forward error correction (FEC) building block," IETF RFC3542, Dec. 2002.
- [3] 佐当 百合野, 西村 祥, 嶋村 昌義, 古閑 宏幸, "誤り訂正技術を用いたTCPレート制御の検討," 電子情報通信学会 技術研究報告, vol. 111, no. 468, NS2011-188, pp. 53-56, 2012年3月
- [4] 佐当 百合野, 古閑 宏幸, 嶋村 昌義, 池永 全志, "誤り訂正技術を用いたTCPの性能改善手法の検討," 電子情報通信学会 総合大会講演論文集, B-6-126, 2013年3月.
- [5] Y. Sohn, J. Hwang, and S. Kang, "Adaptive Packet-Level FEC Algorithm for Improving the Video Quality over IEEE 802.11 Networks," Proc. SERSC International Journal of Software Engineering and its Applications, vol. 6, no. 3, July 2012.
- [6] M. Sakakibara, Y. Tanigawa, and H. Tode, "Highly reliable TCP transfer method with error correction technology," Proc. International Conference on Multimedia, Information Technology and Its Applications (MITA2009), pp. 321-322, Aug. 2009.
- [7] Y. Huang, S. Mehrotra, J. Li, "A hybrid FEC-ARQ protocol for low-delay lossless sequential data streaming," Proc. IEEE International Conference on Multimedia and Expo (ICME2009), pp. 718-725, Jun./Jul. 2009.
- [8] T. Tsugawa, N. Fujita, T. Hama, H. Shimonishi, and T. Murase, "TCP-AFEC: An adaptive FEC code control for end-to-end bandwidth guarantee," Proc. PV2007 International Conference, pp. 294-301, Nov. 2007.
- [9] V. Sharma, K. Ramakrishnan, K. Kar, and S. Kalyanaraman, "Complementing TCP congestion control with forward error correction," Proc. IFIP Networking 2009, pp. 378-391, May 2009.
- [10] L. Baldantoni, H. Lundqvist, and G. Karlsson, "Adaptive end-to-end FEC for improving TCP performance over wireless links," Proc. IEEE International Conference on Communications (ICC2004), pp. 4023-4027, Jun. 2004.
- [11] H. Seferoglu, A. Markopoulou, U. C. Kozat, M. R. Civanlar, and J. Kempf, "Dynamic FEC algorithms for TFRC flows," IEEE Transactions on Multimedia, vol. 12, no. 8, pp. 869-885, Dec. 2010.
- [12] The ns-2 network simulator, <http://www.nsnam.org/>.