Forward Error Correction による損失パケット回復の性能評価

Performance assessment study of loss packets recovery with Forward Error Correction

田沢力　　山内長承
Tsutomu Tazawa　　Nagatsugu Yamanouchi

東邦大学大学院理学研究科情報科学専攻
Toho University　Depertment of Information Sciense

〒274-8510　船橋市三山 2-2-1
TEL:(047)472-1176
email:tazawa@yy.is.sci.toho-u.ac.jp

１.リアルタイム転送

近年、アナログの音声や画像をデジタル化する技術が飛躍的に進歩し、それらを、TCP/IP を用いたインターネットと併用することで、今までに無い通信の有り様が示されつつある。ただ、リアルタイムでの転送という点では、通信基盤の整備が確立されていないという現状もあり、未だ完全には実現されていない。その障害の最たるものは、転送時に生じるパケット損失である。このパケット損失を回復する手段として、TCP[1]で用いられている再送（ARQ：Automatic Repeat Request)のほか、誤り訂正符号技術を用いた前方誤り訂正（FEC：Forward Error Correction）が提案されている[2]。
TCP で用いられる ARQ 法は、損失したパケットをほぼ100%回復する性能を有しているが、ARQ の①損失発生を監視するために、送信に対して受信側からの応答要求をする②パケット損失回復の手段として、必ず再送を適用する、という二つの性質のため、リアルタイム転送やマルチキャスト転送に用いるには不向きである。

リアルタイム転送は、音声・動画等の再生と転送を並行して行うことによって転送開始の待ち時間を減らし、また受信端での大きなバッファリングを不要とするが、受信端でバッファしないために、転送時の遅延変動に弱い。遅延が大きく増加すると、再生のタイミングに対して間に合わなくなり、再生上の欠落が発生する。ARQ による損失パケット回復は、損失発生の通告（TCP では受信端から ACK 応答が返らないことによる損失検出）と当該パケットの再送が必要であり、再送されたパケットでは最低でもパケットの往復時間（RTT：Round Trip Time）程度の余分な遅延が発生する。RTT が比較的長い環境では、パケット損失時に遅延が RTT の分増加することは大きな変動となり、再生上の欠落発生の原因になるし、これを防ぐには受信端で遅延変動を吸収す

るだけのバッファリングを施すことになり、転送開始待ち時間が長くなる。

また、インターネット電話などの対話型のアプリケーションでは、転送遅延の絶対値が小さいことが要求されるが、上述した ARQ の再送処理に伴う RTT 分程度の追加遅延によって、遅延の許容量を超えることが予想される。具体的には、電話で自然な会話が成立するためには往復遅延が 300ms 程度以下である必要があるが、片道の遅延が 100ms 程度の転送路では、通常は往復遅延が 200ms 程度であるが、再送発生時には追加の遅延により 400ms 程度になり、不自然さが目立つようになる。

もう一つの問題であるマルチキャストは、１対多通信であるが、パケット損失の回復手段として ARQ 法を用いると、受信端から送信端に対して何らかの方法でパケットの損失（もしくは到着）を通告しなければならない。多数の受信端に対して通報する場合、通告の数が増えて、戻りの通信路や送信端の負荷が過大なものになる（ACK Implosion 問題)。インターネットの持つ、ネットワーク規模をなるべく制約しない（scalability）という原則に対して、この制約は望ましくない。

ARQ 法の持つこれらの問題を回避する手段として、誤り訂正符号を用いて損失パケットを回復する FEC が提案されている[2]。FEC の原理は、送信端で送信データに冗長符号を付加し、転送路中でパケット損失によって情報の一部が失われても、冗長符号により回復しようというものである。FEC は本質的に、再送の手続きを踏まないので、それによる RTT 分の追加遅延がなく、また受信端から送信端への応答情報が不要なので、応答集中の問題も発生しない。
誤り訂正符号は、従来は半導体メモリや無線通信の雑音による誤り、磁気媒体記録再生誤りなどのビット誤り回復に広く用いられているが、一般に連続した誤り（バースト誤り）の訂正は困難で、誤りバースト長に応じて訂正の計算コストが高くなると言われている。パケット全体が抜け落ちるパケット損失の場合、バースト長がたとえば 1000 バイト程度あり、既存の応用に比較して非常に長いので、誤りを分散させる工夫が必要となる。誤り（損失パケットの場合は損失であるが）の分散には、例えばインターリーブ法[3]がよく用いられる。パケット損失に対して用いる場合、複数のパケットをまとめて１つのブロックとしパケットを縦において並べる。横の行に対して冗長符号を計算して対応する行に置く。最後に冗長符号のための列をパケットとして送出する。これによって、１個のパケットの損失が各冗長計算単位（横の行）では１個の誤り（損失）として扱うことができる。

２.Forward Error Correction の性能モデル

FEC による損失パケット回復の性能は、複数のパケットを単位とする冗長符号処理ブロックの中で、全パケット数 n の中から冗長パケット数 r までの損失に対して回復可能である。全体のブロックで見ると、n パケットからなる処理区間（ブロック）の中で、パケット損失数が冗長パケット数を超えなければ、そのブロックに含まれるもとのデータが回復できるが、それを超えれば回復できない。すなわち、処理区間内にパケット損失がいくつ発生するかによって、回復の可否が決まる。これは、単純に見ると、平均パケット損失発生率 p、ブロック長 n に対して、ブロック内で発生する損失パケット数の期待値は np であるから、冗長パケット数 r を np にとればすべて回復できるように考えられるが、これは正しくない。また、パケット損失のブロック内のパケット数を十分にカバーできるだけの冗長数を加えれば、回復率を 100%にすることもできるが、通信コストの面で、現実的ではない。従って、冗長数と回復性能を論ずるには、ブロック内でのパケット損失発生の分布モデルが必要である。ここでは、ブロック内のパケット損失数の分布を、実測値から分析することを考える。インターネットにおいて、パケット損失の発生は、そのほとんどがネットワーク上で起こる輻輳

という現象に起因すると考えられる。輻輳時のパケット損失は一時的なバッファあふれによるもので、このあふれはある期間の中に集中すると言われている（バースト性）。誤り訂正符号はこのような連続した誤り（バースト誤り）の訂正は困難で、誤りバースト長に応じて訂正計算のコストが非常に高くなると言われている。よってそれを緩和するためには、インターリービング等の手法を用いる必要がある。今回の研究では、バースト性を緩和する手段としてインターリービングを用いた。

３.パケット損失の実測データとパケット損失の傾向分析

パケット損失がどのように起こるかその傾向を見るために、パケット損失の発生状況を測定した。インターネット上の２地点をそれぞれ送信端、受信端と定め、ビデオ等の定ビットレートストリーム送信を模擬するために送信端から等間隔に等サイズのパケットを UDP により転送し、受信端での到着時刻、およびパケット損失状況を測定した（到着時刻は損失モデルには直接関係ないが、時間帯ごとのパケット損失状況を比較するために測定したものである）。送信端、受信端は、SINET を通してつながっている。送信端は学内ネットワークで直接バックボーンに接続され、受信端は 1.5Mbps のアクセス回線でバックボーンに接続されている。転送はパケット長が UDP ペイロード 70 バイト 10mS 間隔で送出したが、これは、約 56kbps の低速ビデオトラフィックを想定したものである。(70byte/10ms=7000byte/s=56kbps)。実測は、転送中に失われたパケットのシーケンス番号を記録することにより、何番目に送信したパケットに損失が起こったかを測定した。送出したパケット数はいずれも 720000（2 時間）である。また、分析に際して、ブロック長を n=10,20,50 として、ブロック内の損失の傾向を調べた。

標本	測定日時		損失パケット総数	到着率	全区間の平均損失数	0 を含まない区間の平均損失数
1	2001/7/23	14:00～16:00	4650	98.64%	0.064	4.745
2	2001/7/24	14:00～16:00	4405	98.72%	0.061	4.762
3	2001/7/24-2	17:00～19:00	2207	99.22%	0.031	3.934
4	2001/7/25	14:00～16:00	10568	96.78%	0.147	4.553
5	2001/7/26	17:00～19:00	20018	94.04%	0.278	4.666
6	2001/7/26-2	20:00～22:00	578	99.67%	0.008	3.658
7	2001/7/27	15:00～17:00	372	99.77%	0.005	3.612
8	2001/7/30	14:00～16:00	28863	91.30%	0.401	4.606

表１　パケット損失発生状況の実測データ（n=10 の場合）

標本	測定日時		損失パケット総数	到着率	全区間の平均損失数	0を含まない区間の平均損失数
1	2001/7/23	14:00～16:00	4650	98.06%	0.194	6.662
2	2001/7/24	14:00～16:00	4405	98.14%	0.184	6.584
3	2001/7/24-2	17:00～19:00	2207	98.78%	0.092	5.436
4	2001/7/25	14:00～16:00	10568	95.41%	0.440	6.401
5	2001/7/26	17:00～19:00	20018	91.64%	0.834	6.651
6	2001/7/26-2	20:00～22:00	578	99.67%	0.024	4.817
7	2001/7/27	15:00～17:00	372	99.77%	0.016	4.482
8	2001/7/30	14:00～16:00	28863	87.63%	1.203	6.482

表２　パケット損失発生状況の実測データ（n=20 の場合）

標本	測定日時		損失パケット総数	到着率	全区間の平均損失数	0を含まない区間の平均損失数
1	2001/7/23	14:00～16:00	4650	96.69%	0.323	9.769
2	2001/7/24	14:00～16:00	4405	96.90%	0.306	9.877
3	2001/7/24-2	17:00～19:00	2207	98.19%	0.153	8.456
4	2001/7/25	14:00～16:00	10568	92.80%	0.734	10.18
5	2001/7/26	17:00～19:00	20018	87.34%	1.390	10.98
6	2001/7/26-2	20:00～22:00	578	99.46%	0.040	7.410
7	2001/7/27	15:00～17:00	372	99.53%	0.026	5.552
8	2001/7/30	14:00～16:00	28863	81.22%	2.004	10.67

表３　パケット損失発生状況の実測データ（n=50 の場合）

到着率とは、720000 個のパケットを例えば n=20 のブロックに分けたとき（全区間数は 720000/20＝36000）にそのブロック内のパケットすべてが到着した数の割合を示しており、ここではブロック内でパケット損失が 0 であるブロック数の割合と一致する。全区間の平均損失数は、各ブロック内の損失数の平均をとったものであり、0 を含まない区間の平均損失数は、全区間のうちで損失数が 0 であるブロックを除いた損失数の平均である。

この実測データから、次の点がわかった。

・すべての標本に共通して、ブロック長によらず損失がまったく無いブロックが 80～99%を占めている。
・パケット落ちが起きているブロックの損失平均が、それぞれの n の値に対しておおよそ一定である(n=10 のときは 4.6～4.7、n=20 では 6.5～6.7 前後、n=50 では 9.8～11.0)。

全ブロックの 90%以上がパケット落ちのまったく無い区間であり、これだけで、パケットの到着率を 90%程度確保できることになる。よって、まず各ブロック内でのパケット落ちがあるか無いかを区別し、パケット落ちがある区間に対して、分析を行う必要性が生じる。

これについて、ブロック内の損失数が 1 個　　得られた。
以上発生している区間を抜き出して分析を
行ったところ、図１～３のようなデータが

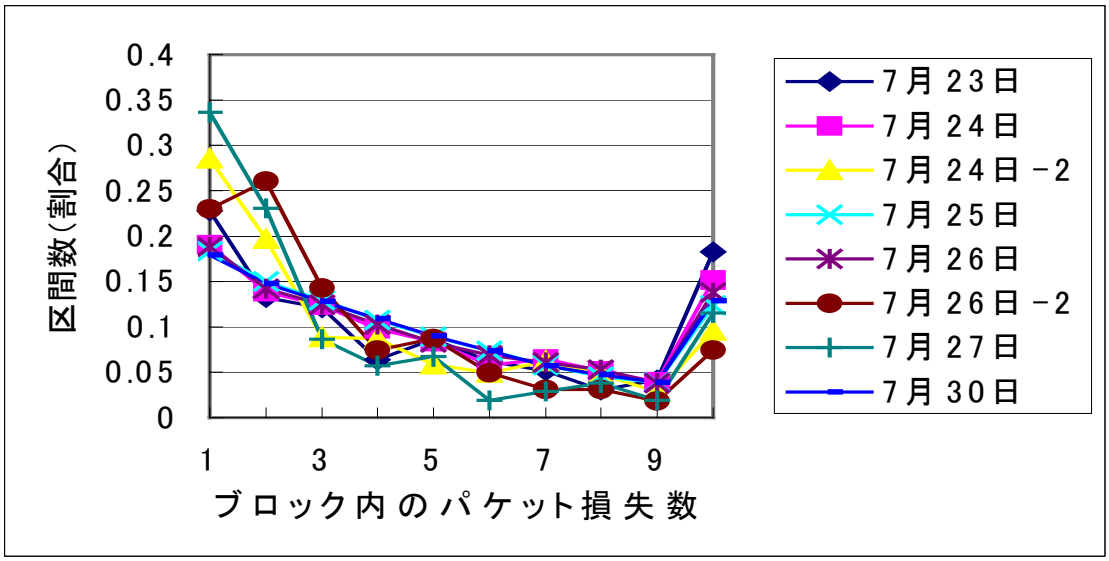

図１　ブロック内のパケット損失数の割合(n=10 の場合)

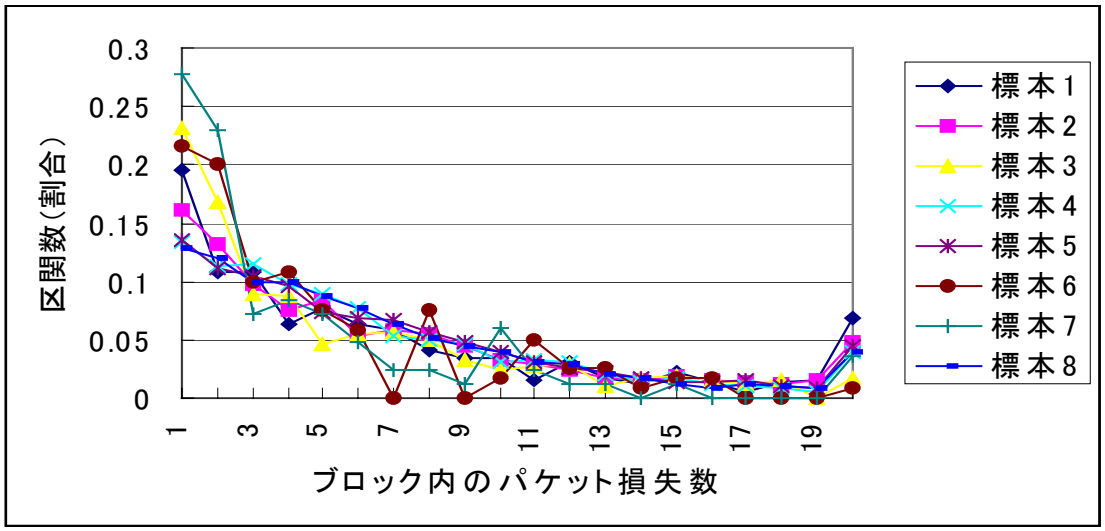

図２　ブロック内のパケット損失数の割合(n=20 の場合)

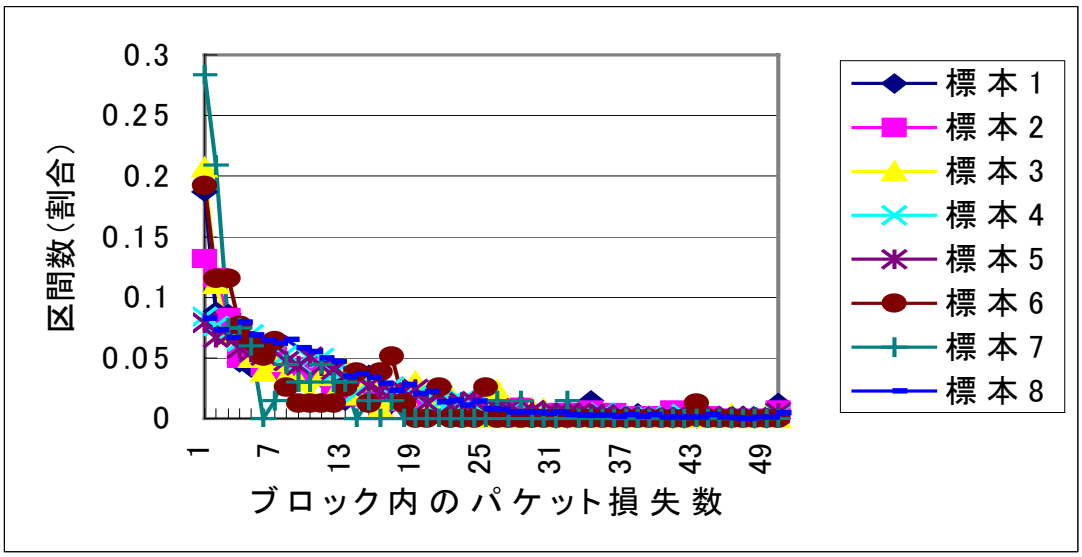

図３　ブロック内のパケット損失数の割合(n=50 の場合)

図から、各ブロック内におけるパケット損失数の割合の分布は、それぞれの区間の長さ n に対して、パケット損失の総数に関わらずほぼ一定に近づくことがわかる。よって、常に一定の冗長度を加えることで、一定の損失パケットの回復性能が得られると推定できる。

４.損失分布からの FEC パケット回復性能の分析

パケット損失の回復はブロックごとに行われ、冗長パケットの数以下の損失数であれば回復でき、それを超えれば回復できないことはすでに述べた。ここでは、先の測定によって得られた実測値について冗長パケット数と損失パケットの回復性能の関係を調べ、全体として到着率（ここでは、損失が 0 であるブロックと損失数が冗長度以下であるブロックの割合の合計）がどれだけ得られるかを分析する。ブロック長を n、冗長パケットを r として、n=20 としたときの冗長パケット数 r の値に対する到着率を表４、図４に示す。

標本	測定日時	r =1	r =2	r =3	r =4	r =5	r =6	r =7	r =8	r =9
1	7/23　14:00～16:00	98.44	98,65	98.86	98.98	99.13	99.25	99.37	99.45	99.51
2	7/24　14:00～16:00	98.44	98.69	98.87	99.01	99.16	99.26	99.37	99.47	99.55
3	7/24-2　17:00～19:00	99.13	99.32	99.42	99.52	99.57	99.63	99.7	99.75	99.79
4	7/25　14:00～16:00	96.03	96.55	97.08	97.53	97.94	98.29	98.53	98.76	98.97
5	7/26　17:00～19:00	92.38	93.72	94.59	95.39	96.00	96.58	97.13	98.61	98.00
6	7/26-2　20:00～22:00	99.74	99.81	99.84	99.88	99.90	99.92	99.92	99.94	99.94
7	7/27　15:00～17:00	99.83	99.89	99.90	99.92	99.94	99.95	99.96	99.96	99.96
8	7/30　14:00～16:00	89.23	90.71	91.95	93.17	94.24	95.20	96.00	96.62	97.17

表４　冗長度 r と到着率との関係（n=20 の場合　各欄の数字は%）

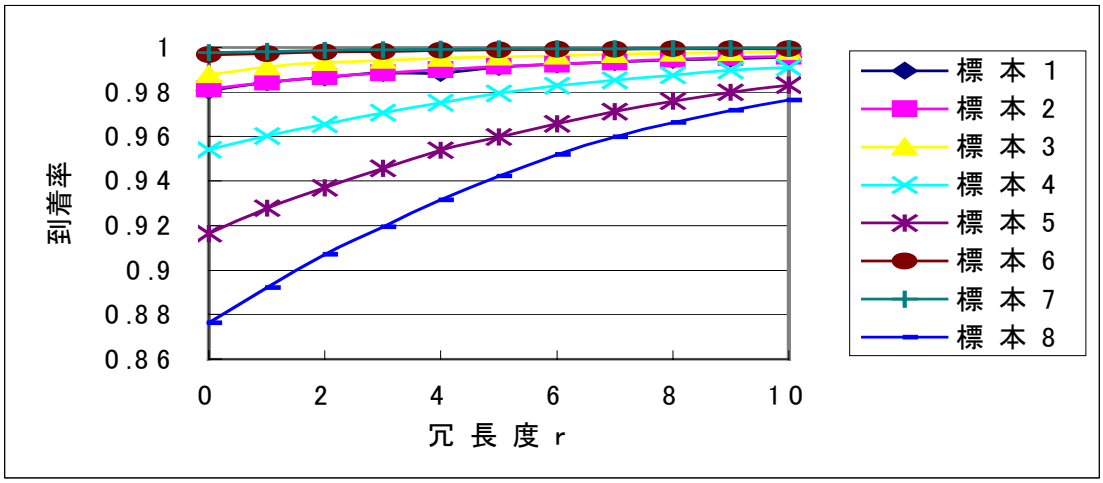

図４　冗長度ｒと到着率との関係

以上のデータから、インターリーブ長がn=20の場合、一定の到着率を得るために与えなければならない冗長度を求めることができる。

５.考察

以上の分析をまとめると、

・インターネット上でのパケット損失状況として、パケット落ちが起こっているブロックの損失数の平均や、それぞれの損失数の割合は、測定した標本の範囲で、回線の混み具合に関わらずほぼ一定であり、FEC を用いて一定のパケット到着率を要求する場合、冗長度ｒはパケットの損失状況によらずインターリービングの区間長によって決まる。パケット損失の回復手段として FEC を用いる場合、例えばn=20とすれば、95%～99%の到着率を確保するために、冗長度ｒを6～9の範囲（n=20 の場合）で設定してやればよい、と言える。
・インターネット上でのパケット損失は、インターリービングのブロック内での損失数の割合が一定であるにも関わらず、損失が0のブロックと0以外のブロックとの数の差が大きいことから、一種のバースト性を有していることが言える。

６.結論

今回、インターネット上におけるリアルタイム転送を実現するために、データ転送で生じるパケット損失の回復に FEC がどれだけの性能を発揮するかを評価した。そのためにはパケットの損失状況を分析することが必要であるが、今回の研究によって、我々の測定した環境下では、その損失状況が回線の状態に関わらず一定であることがわかり、FEC における冗長度を一定の範囲で設定することによって一定の回復性能を実現できることがわかった。
今後の課題としては、低速のビデオとしてはやや不充分な 56kbps のストリームであったので、例えば 128kbps 以上の高速なストリームを仮定してパケット送信を行うことと、パケットの損失状況のモデル化をどのように行えば良いかを検討することである。

参考文献

[1] Postel, J: Transmission ControlProtocol. RFC 793, Internet Society.(1981)
[2] Rizzo, L: Effective Erasure Codes for Reliable Computer communication Protocols. ACM Computer Communication Review, 27-2,pp.24-36.(1997)
[3] 江藤良純、金子敏信:誤り訂正符号とその応用.オーム社.1996