

回線エミュレーションにおける誤り訂正符号の研究

2005MT090 大澤 典久 2006MI172 高木 啓志

指導教員 奥村 康行

1. はじめに

回線エミュレーションは、既存の同期通信サービス(Time Division Multiplexing(TDM))を、非同期通信ネットワーク(Packet Switched Network(PSN))を介して接続する方法である。既に複数の方式が提案され標準化もされているが、それによって、既存TDMとPSNとの統合が可能となり、ネットワークの経済化・柔軟化をもたらすことが期待できる[1]。

しかし現状の回線エミュレーションでは、非同期であるPSN上にクロックなどが同期しているTDMを構築するため、品質保持・劣化抑圧などの技術的課題が存在する。課題のひとつとして、PSNの packet lossに起因するTDMの品質劣化の問題がある。

先行研究[2]においては、TDMの品質劣化に繋がるバーストエラーの問題を抑圧する方式が提案されていた。本研究では、その前段階であるPSNにおけるpacket lossを低減するために、様々な誤り訂正符号の試行・比較を行い、回線エミュレーションにおけるpacket lossをどの程度低減できるかを検証していく。

2. 回線エミュレーションと課題

2.1 回線エミュレーションについて

回線エミュレーションとは、既存のTDM回線間をPSNを介して接続する構成をとっている。図1はその構成を示している。

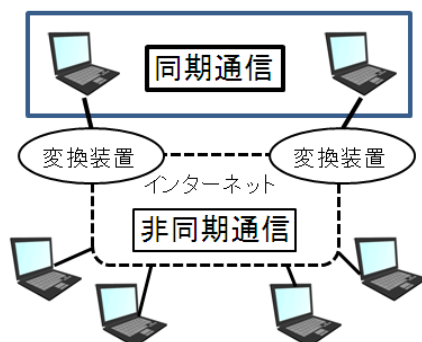


図1 回線エミュレーションの構成

同期通信とは送信側と受信側で同期をとりながらデータ転送を行う通信であり、主に専用線などに用いられる。専用線については次項で解説を行う。一方で、非同期通信とは同期通信のようにクロックのタイミングを一致させなくてもよい通信であり、インターネットもこれにあたる。

2.2 専用線について

2.2.1 専用線が使用されている場合

一般的に、専用線とはある特定の2地点間を結ぶデータ通信専用の回線のことを指すが、二地点間のものだけではなく星型・分岐型の構成も可能である。現在では専用の通信線路や電波周波数帯域を用いずに、他の回線と多重化されているものが主流となっている。

主に公衆網の途絶時も確保しなければならない通信や、改竄・盗聴を防止しなければならない通信のセキュリティを確保するため、警察電話・消防電話・鉄道電話・電力保安通信線・水運用電話などの重要通信で現在では用いられている。他には ATM 放送局のスタジオから送信所へのコンテンツの伝送などがある。

2.2.2 専用線を使用するメリット・デメリット

専用線を使用するメリットとして挙げられるのは、公衆網の輻輳に影響されない(輻輳に起因する事象については後述する。)ことである。また、公衆網よりも機密性が高く盗聴や情報が漏えいすることが少ない。また、99.999%以上の稼働率、故障時には30分以内に復旧作業を完了させるといった高い信頼性を備えたサービスである。

デメリットとして挙げられるのはコストパフォーマンスの悪さである。例として、東京-大阪間を6Mbpsの専用線を用いて通信する場合、月額400万円程度のコストがかかる。

2.3 パケット廃棄に起因するTDMのバーストエラー

回線エミュレーションでは1パケットあたりに多くのビットのTDM信号がカプセル化されている。従って回線エミュレーションでは、PSNにおいてパケット廃棄が発生すると多数のビットのTDM信号が一度に失われ、バーストエラーが発生する。

パケット廃棄のメカニズムとして二つのタイプに分類することができる。

「輻輳型」… パケットスイッチ・ルーターにおいて輻輳が発生した場合に、それらの負荷低減のためにパケットが

廃棄される。

「エラー型」… パケットスイッチ・ルーターにおいてパケット上にエラーが検知された場合に、誤転送を防止するために該当パケットが廃棄される。

この内エラー型パケットロスについては、先行研究[1]において廃棄パケット補償方式の提案・実験的検証がされており、回線エミュレーションにおけるバーストエラーが抑圧することができている。

本研究では輻輳型パケットロスに対して誤り訂正符号を用い、パケットロス率の低減をはかる。

3. ネットワークシミュレーションによるパケットロス率の解析

3.1 ネットワークモデル

図2のようなネットワークポロジータを持つネットワークシミュレータを、NS2を用いて作成する。ノード n0-n3 が専用線、それ以外のノードはインターネットとする。ノード n0-n3 間のプロトコルは UDP を用い、伝送速度は 64Kbps に固定する。パケット廃棄を発生させるためのクロストラヒックとして、50 個から 100 個の TCP をプロトコルとしたネットワークを考える。さらに UDP をプロトコルとしたネットワークを配置する。伝送速度は専用線と同等の 64Kbps とする。

ノード n1-n2 間はバックボーンネットワークとし、帯域を 100Mbps で固定する。

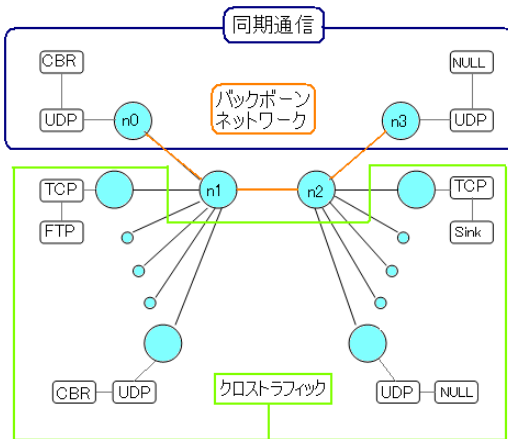


図2 ネットワークポロジータ

3.2 ネットワークシミュレーション

3.2.1 シミュレーション内容

図2のネットワークポロジータを持ったネットワークシミュレータにより、ネットワークシミュレーションを行った。実行時

間は 20 秒とし、実行時間内のパケット行き来をトレースファイル(out.tr)に記録することとする。

3.2.2 実行結果の妥当性

本研究に用いる回線エミュレーションは、参考文献 [3] を参考し作成した。プログラムの妥当性を調べるため、実行結果からスループットを計測した。各スループット(TCP スループット・UDP スループット)の合計スループットがバックボーンネットワークの帯域である 100Mbps を超えていなかったため、今回作成したプログラムには妥当性があると言える。各スループットを計測するプログラム(throughput.pl)は参考文献 [3] より引用したものである。

3.3 パケットロス率の測定

3.3.1 測定内容

3.2.2 により妥当性を確かめたネットワークシミュレーションにおいて、以下の場合のパケットロス率を求める。

- 測定条件
 - (a) クロストラヒック無し
 - (b) TCP クロストラヒック (50 個～100 個)
 - (c) 測定条件 (b) に加えて UDP クロストラヒック
- 伝送速度 (64Kbps)
- 測定対象
 - ノード n0 から n3 へ送られるパケット数
- 測定時間
 - 20 秒

3.3.2 測定方法

3.2 で作成したトレースファイルを参照する。全データの内、パケット到着イベントとパケット廃棄のイベント、送信元アドレス(0.0)とあて先アドレス(3.0)を引数とする。

また、3.2.2 で用いたプログラム(throughput.pl)を改変し、パケット到着数を測定するプログラム(tadr.pl)とパケット廃棄数を測定するプログラム(tadd.pl)を作成する。上記の 4 つの引数と 2 つのプログラムを用いて、パケット到着数・廃棄数を求める。

その後、式(1)に 2 つの数を代入し、パケットロス率を求める。

$$X = Y/Z \quad (1)$$

X : パケットロス率 Y : パケット到着数 Z : パケット廃棄数

求めたパケットロス率を図3に示す。図3において縦軸はパケットロス率、横軸はTCPの個数を表している。

3.3.3 測定結果

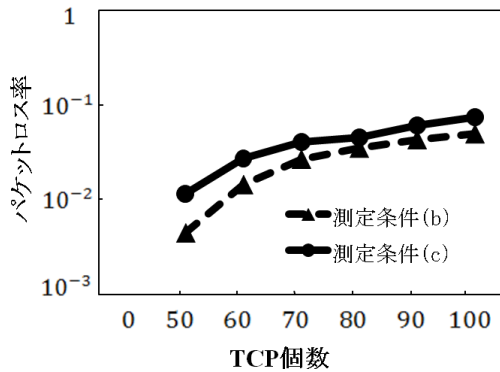


図3 パケットロス率

測定条件(a) (TCPの個数が0)の場合は、パケット廃棄が起こらずパケットロス率は0となった。

測定条件(b)の場合は、TCP クロストラヒックがバックボーンネットワークに付加されたことにより、専用線の通信が阻害されパケット廃棄が起こった。個数が増えるにつれパケットロス率も上昇する結果となったが、個数が増えることで負荷も高くなっていくので正常な結果といえる。また、TCPの個数が45個以下のときにはパケットロス率は0となった。

測定条件(c)の場合、測定条件(b)のときと同様にパケット廃棄が起こり、クロストラヒックの負荷が高くなるにつれてパケットロス率も上昇した。測定条件(b)の場合と比べると、UDPのクロストラヒックが足されているため、パケットロス率は高くなった。また今回の測定ではパケットロス率が比較的高い結果となった。理由としては、研究の対象としたノードn1からn3に送られた総送信パケット数が、20000パケットという少ないパケット数だったことがあげられる。パケットロス率の低減は、パケット廃棄が起こった測定条件(b)と(c)の場合を考えていくこととする。

4. 誤り訂正符号

4.1 パケットロス率の低減

3.3において測定した、パケットロス率を低減化することが本研究の目的である。本研究では、ハミング符号とBCH符号とよばれる誤り訂正符号を用い、各々のパケットロス率がどのように低減するかを検証する。

4.2 誤り訂正符号の機能

誤り訂正符号とはデジタル通信やデジタル記録システムの信頼性を向上させる為のものである。

ハミング符号・BCH符号における誤り(検出)訂正では、k単位長(ビット)の符号を $n = m + k$ 単位長の符号語に変換する。m単位長(ビット)は誤り訂正符号を表す。これを(n, k)符号と呼ぶ。

符号語は最小ハミング距離dが $d > 1$ 、つまり互いに少なくともd単位ごとに異なっている。この冗長性を利用することで誤り検出訂正を行うが、付加されるm長によって、誤りが起きた時に復元できるデータの大きさ(ビット数)は異なる。本研究で用いる符号を表1に示す。表1は参考文献[4]から参照した。

表1 誤り訂正符号(ハミング符号・BCH符号)

誤り訂正符号	符号長	データ	訂正用冗長ビット	誤り訂正可能ビット
ハミング(7,4)符号	7	4	3	1
BCH(15,7)符号	15	7	8	2
BCH(31,11)符号	31	11	20	5
BCH(31,6)符号	31	6	25	7

4.3 誤り訂正符号を用いた復元方法

例としてハミング(7,4)符号を導入した場合のパケットの復元について以下に記す。

送信されるパケットの内、7パケットごとを一つとして考え7×8行列として縦方向に誤り訂正を行う。その結果、通常のハミング(7,4)符号が1つのビットエラーまで復元可能であるため、7つのパケットの内、1パケットの廃棄までなら復元が可能となる。パケットが復元可能な場合の誤り訂正符号の働きとネットワークのモデルは図4に示す。

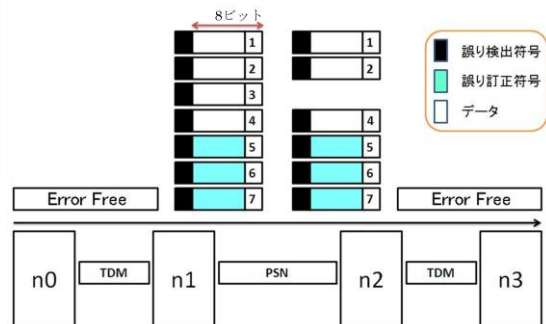


図4 誤り訂正符号を用いた場合の復元モデル

4.4 パケットロス率の低減化

3.3において測定したパケットロス率に対して、表1で示した誤り訂正符号を用いて低減化を目指す。誤り訂正符号の実現には誤り訂正符号ごとに perl プログラムを作成した。3.2で作成したトレースファイルに対して実行し、パケットロス率の低減を行った。実行結果を図5、図6に示す。図5は測定条件(b)に対して、図6は測定条件(c)に対して低減化をはかったものである。

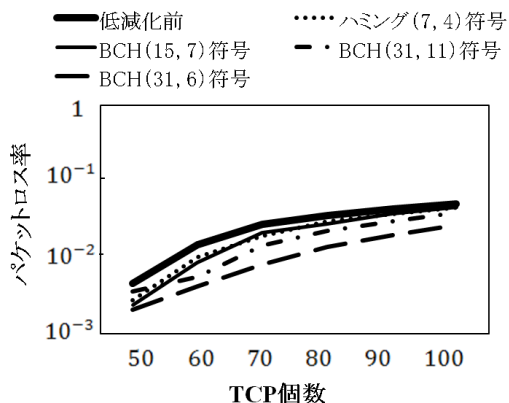


図5 誤り訂正符号によるパケットロス低減化
(TCP クロストラフィックのみ)

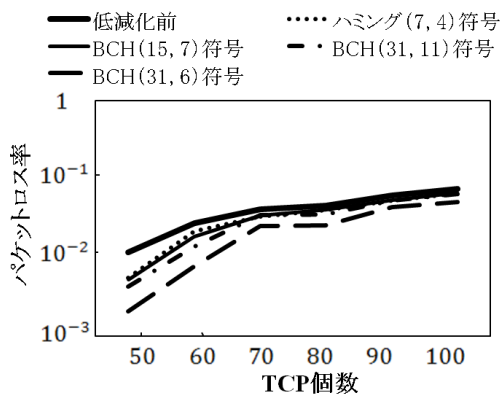


図6 誤り訂正符号によるパケットロス低減化
(TCP クロストラフィック+UDP クロストラフィック)

4.5 考察

どちらも場合も、全ての誤り訂正符号によってパケットロス率の低減化は起こった。誤り訂正符号の内、ハミング(7,4)符号と BCH(15,7)符号による低減率がほぼ等しい結果となったが、これは誤り訂正可能ビットがあまり変わらず、そして訂正能力が低いからだと考えられる。一方で BCH(31,6)符号は訂正可能ビット数が大きいため最もパケットロス率が低減した。

TCP クロストラフィックのみの場合、TCP の個数が 60~70 個のときに誤り訂正符号ごとの低減率の差が大きくなった。ハミング(7,4)符号では 0.3355、BCH(31,6)符号においては 0.6639 となった。TCP の個数が増大するにつれて低減率が減少する傾向が見られたが、これはクロストラフィックの増大につれてパケットロスが連続的に起こったからだと考えられる。

さらに UDP クロストラフィックを追加した場合、TCP が 50 個のときに最も低減率に差がみられた。しかし TCP が

100 個になると、ハミング(7,4)符号・BCH(15,7)符号・BCH(31,11)符号において低減率に差が少なくなり、低減率は約 0.101 となった。訂正可能ビット数が 5 である BCH(31,11)符号と、訂正可能ビット数が 1 であるハミング符号の低減率にあまり差が見られないのは、クロストラフィックの増加による連続したパケットロスに対して訂正が追いつかなかったためであるが、クロストラフィックが TCP のみのときよりも顕著な結果となった。そのような状態でもパケットロス率を大幅に低減させるには、さらに強力でビット訂正できる誤り訂正符号が必要となってくるであろう。

5. まとめと今後の課題

本研究では、TCP と UDP の混在する回線エミュレーションを作成し、ネットワーク上のシミュレーションにおいて発生するパケットロス率を計測した。実際には様々なクロストラフィックによる負荷をかけた場合にパケットロス率がどのように変化(クロストラフィックによる輻輳のパケットロス率の変化)するかを計測した。その後、得たパケットロス率を対象として様々な誤り訂正符号(ハミング符号・BCH 符号)を用い、それぞれのパケットロス率の低減率を調べた。

検証した結果、やはり誤り訂正可能ビット数の多い符号の方がパケットロス率の低減率は大きいことが分かった。しかし誤り訂正可能ビットが異なっていたとしても、パケットロス率がさほど変わらないような結果も得ることができた。今後の課題としては、そのような場合にどんな誤り訂正符号がさらにパケットロス率の低減につながるかを、より多くの誤り訂正符号を比較・検証することで求めていく必要がある。

またパケットロス率の大きさによってどこまでの訂正能力をもった誤り訂正符号を使用すると最も効率が良いのかなどの検証も行っていくかなければならない。

参考文献

- [1] 深田 陽一, 斎藤 幸一, 前田 洋一, "[招待講演] 回線エミュレーションの技術動向," 電子情報通信学会, 信学技報 CS2005-77, 2006-1.
- [2] 深田 陽一, 斎藤 幸一, 前田 洋一, "回線エミュレーションにおける TDM 信号のバーストエラー抑圧のための破棄パケット補償技術の提案と実証," 電子情報通信学会, 信学技報 CS2006-16, 2006-5.
- [3] 銭 飛, "NS2 によるネットワークシミュレーション," 森北出版株式会社, 2007-10.
- [4] 牧野 伸一, "誤り訂正符号について(ハミング符号, BCH 符号, RS 符号)," 名古屋工業大学電気情報工学科 岩波研究室, 2004.
http://iwanami-web.elcom.nitech.ac.jp/member/h13sotsuken/block_code.pdf