

LT 符号を用いたマルチキャストコンテンツ配信における 追加要求方式の検討

佐々木 力[†] 小林 修^{††} 長谷川輝之[†] 阿野 茂浩[†] 長谷川 亨[†]

[†] (株) KDDI 研究所 〒 356-8502 ふじみ野市大原 2-1-15

^{††} KDDI 株式会社 〒 102-8460 千代田区飯田橋 3-10-10

E-mail: [†]ch-sasaki@kddilabs.jp

あらまし IP マルチキャストを用いたコンテンツ配信サービスでは、信頼性向上の目的で FEC を適用する形態が一般的である。特に、即時性が要求されない蓄積型の配信に適した FEC として LT 符号が知られている。LT 符号を用いると配信時間拡大に応じて信頼性が向上するが、現実的には配信時間には制限がある。一方、受信側では、途中で受信を開始し十分な配信時間を確保できない、あるいは、モバイル環境等で通信品質が想定より劣悪である等の理由により、配信未了者が発生する可能性もある。本稿では、LT 符号を用いたマルチキャスト配信後の処理として、配信未了者に対する効率的なユニキャストベースの配信完了保証方式を提案し、シミュレーションにより提案方式の有効性を検証する。

Add-on Request Method for Multicast Content Distribution using LT codes

Chikara SASAKI[†], Osamu KOBAYASHI^{††}, Teruyuki HASEGAWA[†], Shigehiro ANO[†],
and Toru HASEGAWA[†]

[†] KDDI R&D Laboratories Inc., 2-1-15 Ohara, Fujimino-shi, 356-8502 Japan

^{††} KDDI Corporation, 3-10-10 Idabashi, Chiyoda-ku, 102-8460 Japan

E-mail: [†]ch-sasaki@kddilabs.jp

Abstract Recently erasure codes based on XOR operation have become major FEC candidates. In particular, LT codes are effective in a multicast stored data download because these codes can achieve higher reliability by expanding the receiving time at each client with heterogeneous communication quality. However, multicast transmission time can be restricted. In order to assure the download completion, a unicast based fail-safe download scheme after multicast distribution is required. We newly proposed an add-on request method by the download incompleteness receiver in this scheme considering LT codes characteristics. We confirmed the efficiency of our proposed method by simulation.

1. ま え が き

近年、固定・モバイルアクセス環境のブロードバンド化が進展し、大容量コンテンツの一斉同報など、放送型サービスへの要望が高まっている。このようなサービス

においては、ネットワーク帯域を有効利用できる IP マルチキャストの適用が考えられる。一方、高信頼なマルチキャスト配信を実現するためには、パケットロスに伴うデータ損失に対して、高信頼化技術の適用が必要となる。高信頼化技術としては再送 (ARQ: Automatic Repeat

reQuest)[1], [2] や配信データの冗長化 (FEC: Forward Error Correction)[3], [4] などが挙げられるが、非常に多くの受信者数が想定される放送型サービスでは、ARQ を用いると再送サーバへの応答爆発が問題となるため、FEC を用いるのが一般的である。

パケットロスを回復する FEC としては、XOR 演算ベースの erasure 符号が有効である [5] ~ [8]。これらの符号は、高速な符号化・復号化が可能で、また優れた誤り訂正能力を有している。特に、半永久的符号化が可能な LT 符号 [6] を用いると、送信者はマルチキャスト配信を継続する一方、各受信者では受信時間を拡大することでより高い信頼性を実現することができる。そのため、コンテンツの同報など即時性が要求されない蓄積型の配信サービスに適している。また、IP 放送のようなストリーミング型の配信サービスにおいても、即時性が必要な映像・音声データ以外にも、番組情報・番組表・付加映像などの即時性の少ない蓄積型の同報データは存在し、その配信に適用できる。

LT 符号を用いたマルチキャスト配信では、受信者が、配信され続けている符号化データの中から自身の都合 (受信開始時刻や通信品質) に合わせて受信時間を決定し復号する配信形態が可能である。一方で、中途での受信開始や想定外の通信品質により配信完了が非常に遅れることは、コンテンツの価値を減ずることになり、また、時期を逸したコンテンツに対する受信要求は少ない。特に、モバイル端末を対象とした場合、無線帯域等の限られたリソースの有効利用の観点から、このような少数の受信者のためにマルチキャスト配信時間を無制限に延長することは現実的ではない。従って LT 符号を用いた場合も、スケジュールに従ってマルチキャスト配信時間を固定にするサービス運用シナリオが想定される。本シナリオでは、マルチキャスト配信終了時点で配信が完了していない受信者 (配信未了者) が発生する可能性がある。

そこで本稿では、配信未了者が少数であることに着目し、マルチキャスト配信終了後に、TCP を用いたユニキャスト通信により配信完了保証を行う方式について検討する。具体的には、LT 符号の特性に基づき、より少ないデータ転送で配信が完了するように、個々の配信未了者が受信・復号状況に応じて適切な要求データを選択する効率的な手法を提案し、提案手法の有効性を計算機シミュレーションにより明らかにする。

本稿の構成は以下の通りである。2. で一般的な FEC の概要ならびに LT 符号の説明を行う。3. でマルチキャスト配信未了者が行う要求データの選択手法を提案し、4. で他の比較手法も含めた計算機シミュレーションによる性能評価を行う。5. で結果を考察し 6. でまとめを行う。

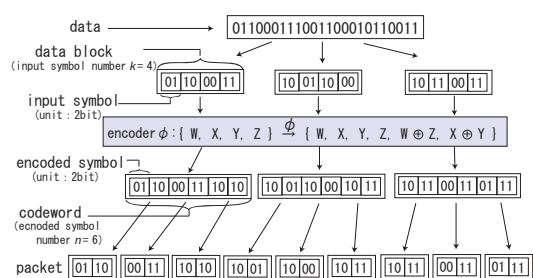


図 1 FEC の処理の流れ

Fig. 1 FEC processing flow.

2. パケットレベル FEC

本節では、パケットレベル FEC の概要、ならびに、本稿の検討対象である LT 符号について説明する。

2.1 FEC の概要

図 1 に FEC 処理の流れを示す。送信者 (エンコーダ) はデータを同一サイズのデータブロックに分割し、データブロックをさらにシンボルに分割する。シンボルはエンコーダ ϕ の入出力の最小単位であり、入力側を入力シンボル、出力側を符号化シンボルと呼ぶ。シンボルの単位は、ビット・バイト・パケットサイズなどであり、本稿では、入力シンボルと符号化シンボルのサイズ (単位) はともにパケットのデータサイズとする。次に、送信者は付加する冗長量 (符号化レート) に応じた誤り訂正符号を用いて各データブロックを符号語に変換し、生成された符号化語をパケットに分けて送信する。ここで、データブロック内の入力シンボル数を k 、符号語内の符号化シンボル数を n とすると、符号化レート R は下式で表される。

$$R = \frac{k}{n}$$

一方、受信者は通信路上でパケットロスが発生したとしても、全てのデータブロックを正しく復号できれば、データを回復することができる。データブロックを正しく復号するためには、対応する符号語内の符号化シンボル数が一定以上となるようにパケットを受信する必要がある。Reed-Solomon 符号の場合、入力シンボル数と同数の符号化シンボルを受信すれば (n 個中 k 個)、データブロックの復号が可能である。他方、LT 符号では入力シンボル数より冗長に ($k + \alpha$ 個) 符号化シンボルを受信する必要がある。

一般的に、同一符号化レート R でより優れた誤り訂正能力を持つ符号を構成するためには、入力シンボル数 k (もしくは符号化シンボル数 n) を増大させる必要がある [10]。ところが、Reed-Solomon 符号等のほとんどの符号は計算量 (特に復号) の増加が著しく符号構成も複

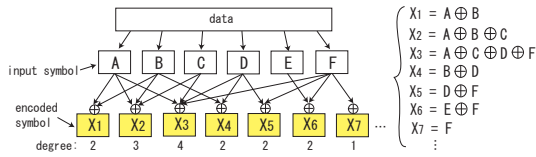


図 2 LT 符号の符号化例
Fig. 2 LT codes encoding.

雑になる．したがって現実的には k の増大には限界がある．近年，以上のような問題点を踏まえ，XOR 演算を用いて符号化・復号化の計算量を削減した *erasure codes* が提案されている [5] ~ [8]．これらの符号化では，入力シンボルそのものか，2 個以上の入力シンボルを XOR 演算した結果を符号化シンボルとする．復号法は符号化に応じた効率的な手法が提案されており，LT 符号では 2.2 節の復号法が使われる．

2.2 LT 符号

Reed-Solomon 符号や LDPC 符号等の一般的な符号では，適切な符号化レートと復号成功率を有する符号を構成するために，送信者は通信路のペケットロス率を予め推定する必要がある．しかしながら，マルチキャスト環境では受信者数が多く，またその通信品質は多様になるためペケットロス率の推定が困難である．また，符号化レートを予め決定し固定的に符号を構成するため，通信品質が想定と異なる場合は，これに適した符号化レートの符号を再構成するか，予め，より小さな符号化レート ($n \rightarrow$ 大) で冗長に生成したペケットを間引いて (パンクチャ [11]) 送信する必要がある．パンクチャにより生成された符号は，適切な符号化レートで生成した符号よりも訂正能力が悪くなる傾向がある．

これに対し LT 符号は，確率的符号化により半永久的 ($n \rightarrow \infty$) な符号化シンボルの生成が可能で，任意のパンクチャを適用した場合も，優れた誤り訂正能力を高確率で提供できるという特徴がある．すなわち LT 符号では，送信者はペケットロス率によらず同一の符号で送信を継続し，各受信者は自分の通信品質に応じて受信時間を変えれば良い．このような性質を有する符号をレートレス符号と呼ぶ．

図 2 に LT 符号の符号化の例を示す．送信者が行う符号化手順は以下の通りである．

LT 符号の符号化手順：

- Step 1: データを k 個の入力シンボルに分割する．
- Step 2: 定義域が $1 \sim k$ の確率分布 $\rho(d)$ に従い， d をランダムに選ぶ．
- Step 3: k 個の入力シンボルから d 個の入力シンボルを一様分布に従って選び，それらを XOR

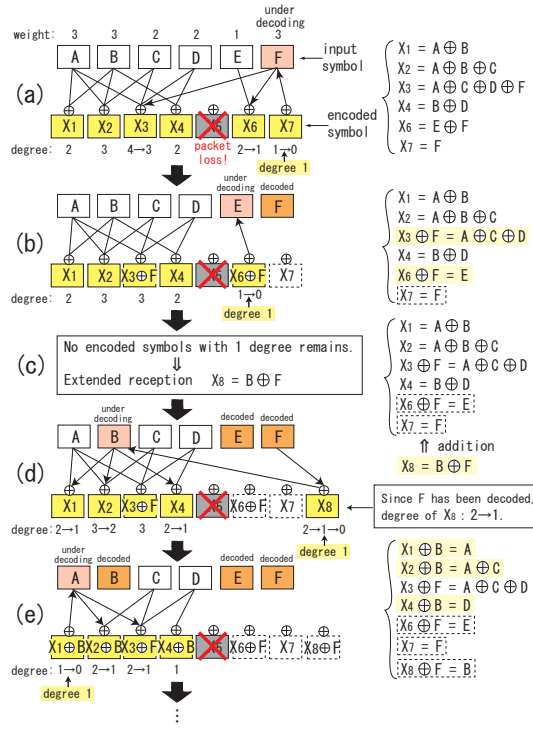


図 3 LT 符号の復号化

Fig. 3 LT codes decoding and extended reception.

演算した結果を符号化シンボルとして送信する．

Step 4: Step 2 ~ 3 を任意の回数繰り返す．

□

LT 符号の性能は，入力シンボル数 k と確率分布 $\rho(d)$ に大きく依存する． $\rho(d)$ としては，*Robust Soliton* 分布 [6] を用いるのが一般的であり，その特性は 2 つのパラメータ c, δ により決定される．以降，ある符号化シンボルを構成する入力シンボル数をその符号化シンボルの次数 (*degree*) と呼ぶ．また，入力シンボルと符号化シンボルの XOR 演算対応の理解に図 2 のようなグラフを用いるため，直接の XOR 演算関係 (被 XOR 演算関係) がある場合その入力シンボルと符号化シンボルは接続してると呼ぶ．なお，次数は，接続入力シンボル数と等しい．図 2 において， X_1 は次数 2 で A, B と接続している．また， A は X_1, X_2, X_3 と接続している．

受信者は，一定数の符号化シンボルを受信後，復号および追加受信を行う．図 3 に復号化の例を示す．図 3 の受信者は，図 2 で生成された符号化シンボルのうち X_5 をロスしている．具体的手順はを以下に示す．

LT 符号の復号化手順：

- Step 1: 受信符号化シンボルの中から，次数 1 の符号化シンボル (X_7) を探す．これは，入力シン

ボル (F) 自身であるため、直ちに復号できる (図 3 (a)).

- Step 2: さらに, Step 1 で復号された入力シンボル (F) を XOR 演算対象とする符号化シンボル (X_3, X_6) に対し, 復号された入力シンボル (F) の値を XOR 演算する. その結果, 当該符号化シンボル (X_3, X_6) の次数は 1 だけ小さくなる (図 3 (a)).
- Step 3: Step 1~2 を繰り返し, 全入力シンボルが復号できれば復号成功となる.
- Step 4: 復号途中で次数 1 の符号化シンボルが見つからない場合は, さらに符号化シンボル (X_8) を受信し Step 1 に戻る (図 3 (c), (d)).
-

以降, ある入力シンボルを XOR 演算対象とする受信符号化シンボル数をその入力シンボルの重み (*weight*) と呼ぶ (図 3 (a) 参照). 次数と同様に, 重みは接続符号化シンボル数と等しい.

LT 符号では, 送信され続けている符号化シンボルの中から入力シンボル数 $k + \alpha$ 個程度を受信すれば高確率で復号に成功する. さらに受信符号化シンボル数を増やすことで復号成功率は向上する. また, 各符号化シンボルが独立に生成されるため, 復号成功率は受信数に依存し受信パターンの影響を受けにくい. つまり, 任意の通信環境に対して, 受信数さえ十分に確保できれば, 高確率でパケットロスを回復できる. また, 受信符号化シンボル数 r に対して, 受信冗長率 $R_r = r/k$ としたとき, LT 符号では, 同じ復号成功率を得るために必要な受信冗長率 R_r は, k を増加させることで 1 に近づく [6]. すなわち, Reed-Solomon 符号と比較して, α/k 程度冗長に受信しなければならないが, 高速な復号が可能であるため k に比較的大きな値を使用できる. したがって, 少ない受信冗長率で優れた訂正能力を得ることが可能となる.

3. 提案手法

マルチキャストを用いて効率的にコンテンツ配信を行うためには, 受信者が同時に受信可能状態になっている必要がある. 放送のモデルはまさにこれに当たる. 番組情報・番組表・付加映像などの蓄積データを LT 符号を用いて配信する場合, 固定配信時間を拡大すれば信頼性も向上する. 一方, 受信符号化シンボルの不適切な組み合わせや, 途中受信による受信符号化シンボルの不足といった理由により, マルチキャスト配信終了時に一部の受信者で配信未了となる可能性がある. しかしながら, これら少数の受信者への配信を完了するために固定配信時間を拡大することは非効率である. そこで, LT 符号を

用いた固定時間のマルチキャストによりほぼ全ての受信者への配信を完了し, 配信未了者はさらにユニキャストで差分データを要求する保証方式を検討する. 保証方式を含むコンテンツ配信の流れを以下に示す.

マルチキャスト配信フェーズ:

- Step 1: 送信者は LT 符号を用いてコンテンツを冗長化し, マルチキャストで送信する.
- Step 2: 受信者は復号によってコンテンツが得られるまで受信を続ける.
- Step 3: 送信者は規定数 s の符号化シンボルを送信後, マルチキャストを停止する.

ユニキャスト保証フェーズ (保証方式):

- Step 4: 配信が完了していない受信者 (配信未了者) は差分データを要求する (要求データ選択).
- Step 5: 送信者は要求データを当該配信未了者へ追加転送する. 必要ならば要求データを生成する.
- Step 6: 当該配信未了者は, 要求データを受信する. 受信後必要であれば復号を行う.
- Step 7: 配信が完了するまで Step4~6 を繰り返す.

以下では, 上記保証方式において, 配信完了までに必要なユニキャスト要求回数 (追加転送回数) を削減するために, Step4 における効率的な要求データの選択手法を提案する. まず, Step6 における復号方針について検討し, Step4 における選択方針 1~3 を示す.

復号方針:

要求データ受信後, 復号を行うこと. □

マルチキャスト配信終了時, 配信未了者は, 復号途中の次数 2 以上の符号化シンボルを比較的多数保持し, 未復号の入力シンボルが多いことが予想される. そこで, 全ての未復号入力シンボルを要求するのではなく, 要求データ受信後に保持している符号化シンボルを利用した復号を行うことで要求回数を抑制できると考えられる. 他方, LT 符号の復号法に代わり, 一般的な連立方程式の解法であるガウスの消去法を用いることで復号が進む可能性もある. しかしながら, 多数の未復号入力シンボルが保持されている場合, 計算量も多い. 本稿では, 計算能力が小さいモバイル端末の存在や複数の復号法実装のコストを考慮し, 処理負荷の少ない LT 符号の復号法のみを用いることにする.

選択方針 1:

入力シンボルを要求すること. □

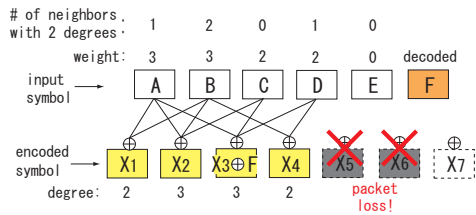


図 4 提案要求方式の例

Fig. 4 A example of proposed mechanism.

要求するデータ候補としては、

- (1) 符号化シンボル
- (2) (未復号)入力シンボル

の 2 通りが考えられる。一方、受信者が真に必要なデータは符号化シンボルではなく入力シンボルであるということに着目すると、未復号の入力シンボルを要求する方が効率的である。

選択方針 2:

重みの大きな入力シンボルを要求すること。 □

本方針は、上記復号方針と選択方針 1 に従うことを前提にする。復号の難しさは保持している符号化シンボルの総 XOR 演算数と関係があると考えられる。そうであるなら、復号をより進める要求入力シンボルは、全体的に符号化シンボルの XOR 演算数を削減するものである。特に、要求入力シンボル受信後の復号で最初に行う代入のみを対象とすれば、失敗時の復号状態において符号化シンボルの XOR 演算構成シンボルとして多く選ばれている(重みの大きな)入力シンボルが有効であると考えられる。

選択方針 3:

次数 2 の符号化シンボルに接続している入力シンボルを要求すること。 □

本方針も、同様に上記復号方針と選択方針 1 に従うことが前提である。復号法を考慮すると、要求入力シンボルの代入により符号化シンボルの次数が 2 から 1 に減少した場合、その次数 1 の符号化シンボルに接続している残り 1 つの入力シンボルが復号される。このように、次数 2 の符号化シンボルの XOR 演算構成シンボルとして選ばれている 2 つの入力シンボルのうち片方を要求受信するともう片方は復号できる。したがって、このような入力シンボルを要求すると少なくとも 2 つの入力シンボルの値が判明する。以上より、復号促進が期待される次数 2 の符号化シンボルに接続している入力シンボルを要求すれば復号が促進されることが考えられる。

上記選択方針 1~3 を考慮して、我々は以下のような 3 種類の新しい要求データ選択手法を提案する。以降、図 4 に従って各手法を説明する。図 4 では、マルチキャスト配信において符号化シンボル X_5, X_6 をロスしたため復号に失敗している (F は復号済み)。

提案手法 (3 種類):

WTmax 手法 (復号方針, 選択方針 1, 2)

- 最大重みの未復号入力シンボル (A or B) を選択する。 □

DG2rand 手法 (復号方針, 選択方針 1, 3)

- 重み 0 の未復号入力シンボルがある場合は、それら未復号入力シンボル (E) を選択する。
 - 最小次数 (= 2) の符号化シンボルに接続している未復号入力シンボル (A or B) をランダムに選択する。 □

DG2max 手法 (復号方針, 選択方針 1~3)

- 重み 0 の未復号入力シンボルがある場合は、それら未復号入力シンボル (E) を選択する。
 - 最も多く最小次数 (= 2) の符号化シンボルに接続している未復号入力シンボル (B) を選択する。 □

$DG2rand$ 手法と $DG2max$ 手法を用いた場合、必ずしも次数 2 の符号化シンボルが存在するとは限らないので対象次数は最小次数とし、重み 0 の入力シンボル (E) が存在する場合は、最初に要求する。なお、 $DG2max$ 手法によって選ばれた入力シンボルは、他の入力シンボルよりも大きな重みになっている可能性は高いため、 $DG2max$ 手法は選択方針 2 に従っている。

4. 性能評価

本節では、要求データ選択手法を LT 符号に適用した場合の性能を計算機シミュレーションにより評価する。

4.1 他選択手法との比較

復号方針と選択方針 1~3 の妥当性を確認するために、以下 3 種類の選択手法についてもシミュレーションを行う。

比較手法 (3 種類)

LTC 手法 (復号方針)

- 送信者で新たに生成した符号化シンボル (X_7 or X_8 or ..., 図 4) を選択する。 □

all 手法 (選択方針 1)

- 未復号入力シンボル ($A \sim E$, 図 4) を全て選択する。 □

random 手法 [13](復号方針, 選択方針 1)

- ランダムに未復号入力シンボル (A, B, C, D , or E , 図 4) を選択する。 □

LTC手法で送信者が生成する要求符号化シンボルはマルチキャスト配信時と同一の次数分布 $\rho(d)$ を用いることとする。なお、LTC手法以外に符号化シンボルを選択する手法としては、*loss* 手法^(注1) が考えられる。しかしながら、ロス個数を超えて要求することがない場合、どちらの符号化シンボルも同一分布から独立に生成されるため、両手法の平均要求回数は同じになる。したがって、ロス符号化シンボルの検出が不要な点で優れている LTC手法のみ評価する。*all* 手法では、一度に全ての未復号入力シンボルを要求するため、復号方針に従う必要はない(3.の保証手法 Step6 は省略)。各方針の妥当性を検証するには、例えば、復号方針の場合、*random* 手法と *all* 手法を比較すれば良い。また、選択方針 1~3 の場合は、*random* 手法とそれぞれ、LTC手法、*WTmax* 手法、*DG2rand* 手法、と比較すれば良い。

4.2 シミュレーション条件

LT 符号では、個々の符号化シンボルが独立かつ同一確率分布により生成されているので、配信未了状態は、マルチキャスト送信符号化シンボル数 s やロスパターンではなく、受信符号化シンボル数 m に依存している [6]。すなわち、転送数、ロス個数、ロスパターンが異なっても、ある受信者一人を対象にすると、受信符号化シンボル数 m が同じであれば、保証方式を用いたときの平均ユニキャスト要求回数は同じになる。そこで、シミュレーションでは、パケットロスは発生しないものとして、 m と s を同じにし、マルチキャスト配信で m 個受信して配信に失敗した受信に対して、各要求データ選択手法を用いた保証方式を行った場合の平均ユニキャスト要求回数を評価する。なお、LT 符号のパラメータは、100 万人規模を対象とした(復号誤り確率 10^{-6} を少ない受信冗長率 R_r で達成する)ものを用いる [9]。シミュレーションの前提条件は、以下の通りである。

- (1) LT 符号のパラメータ
 - (a) $k = 1,000, c = 0.07, \delta = 0.05$
 - (b) $k = 10,000, c = 0.03, \delta = 0.01$
- (2) 要求や転送は、1 送受信あたり 1 入力シンボルもしくは 1 符号化シンボル単位で行う。
- (3) パケットロスは発生しない。
- (4) 符号化の擬似乱数系列は、 2^{32} 周期のメルセンヌ・ツイスタ乱数 [12] により生成する。

4.3 結果

シミュレーションにより得られた各手法別の平均要求

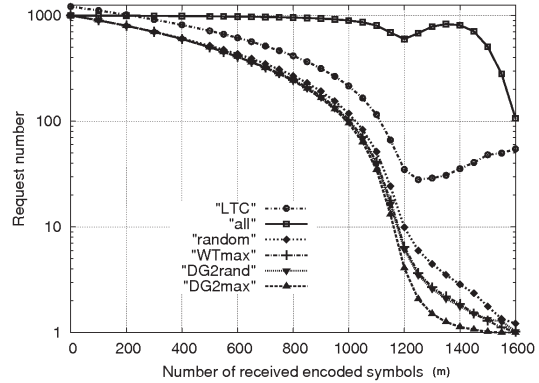


図 5 手法別平均要求回数 ($k = 1,000$)

Fig. 5 Average request number ($k = 1,000$).

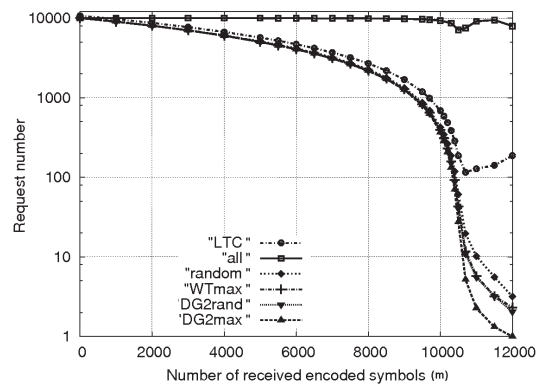


図 6 手法別平均要求回数 ($k = 10,000$)

Fig. 6 Average request number ($k = 10,000$).

回数を図 5, 6 に示す。各手法の平均要求回数の大小関係は以下の通りである。

$$\begin{aligned} all \geq LTC = loss \geq random \\ \geq WTmax \geq () DG2rand \geq DG2max . \quad (1) \end{aligned}$$

ただし、受信符号化シンボル数 m が非常に少ない場合 ($m > 200$, 図 5; $m < 1,000$, 図 6) は、 $all < LTC$ である。

例えば、図 5 の $m = 900$ において、要求回数は、*all* 手法 936 回、*LTC* 手法 316 回、*random* 手法 192 回、*WTmax* 手法 173 回、*DG2rand* 手法 169 回、*DG2max* 手法 166 回、である。不等式 1 より、*DG2max* 手法が最も性能が良いことが分かる。

5. 考 察

5.1 復号方針と選択方針の妥当性

all 手法の平均要求回数は、マルチキャスト終了時の配信未了者が保持する平均未復号入力シンボル数と等しい。例えば、 $k = 1,000$ の場合、 $m = 1,200$ の配信未了者は平均 599 個の未復号入力シンボルを保持している。この

(注1): *loss* 手法 (復号方針)

- ロスした符号化シンボル (X_5 or X_6 , 図 4) を選択する。 □

ように、復号に失敗した場合、多数の未復号入力シンボルを保持しているため、*all*手法を用いると *random*手法と比べ要求回数が非常に多くなる。そのため、要求データ受信後に、保持している符号化シンボルを活用した復号を行うこと（復号方針）は効率的であることが分かる。また、不等式 1 における *LTC*手法と *random*手法を比較することで、要求データは符号化シンボルより入力シンボルの方が効果がある（選択方針 1）ことが分かる。また、*random*手法と *WTmax*手法の比較、および *random*手法と *DG2rand*手法の比較を行うことで、それぞれ、重みの大きな入力シンボル（選択方針 2）が効果的であること、また次数 2 の符号化シンボルに接続している入力シンボル（選択方針 3）が効果的であることが分かる。さらに、*DG2max*手法と他の方式を比較することで、各方針を組み合わせることでより大きな効果が得られることが分かる。また選択方針 2, 3 が有効であることから、配信完了までに複数回要求する場合でも、各要求毎にできるだけ復号を進める入力シンボルを要求すれば十分であることが分かる（複数回の受信後に最も復号が進むような入力シンボル集合を考慮しなくてもよい）。

5.2 選択手法の最適性

より有効な選択手法の存在を確かめるために、各手法の平均要求回数とその理論的な下界値からの差を検証する。なお、ここで扱う下界値とは、最適手法を用いた場合の要求回数がその値よりも小さくならないという値である。マルチキャストでの受信符号化シンボル数を m とすると、下界値 $l(m)$ は、後述の理由により以下のように導出した。

$$l(m) = \begin{cases} k - m & \text{for } m < k \\ 1 & \text{for } m \geq k \end{cases} \quad (2)$$

k シンボルの情報を伝達するには、少なくとも k シンボル受信する必要があるため [10]、 $m < k$ の場合、配信完了まで少なくとも残り $(k - m)$ シンボルが必要である。一方 $m \geq k$ の場合は、 k シンボル以上受信して復号に失敗しているため、要求回数の下界値は 1 である。

図 7, 8 に平均要求回数とその下界値との差を示す。受信符号化シンボル数が非常に多い ($m > 1,400$, 図 7; $m > 12,000$, 図 8) 場合は、*random*手法を用いても数回程度の要求で終了し、下界値に非常に近づいている。また、非常に少ない ($m < 300$, 図 7; $m < 3,000$, 図 8) 場合は、各方式の差はほとんどなく、同様に下界値に近づいている。なお、下界値との差が 0 であるということは、その領域における選択手法の最適性と下界値の厳密性の 2 つを示している。すなわち、受信符号化シンボル数 m が非常に多い場合や少ない場合は、*random*手法でもほとんど最適であることが分かる。一方、受信符号化シ

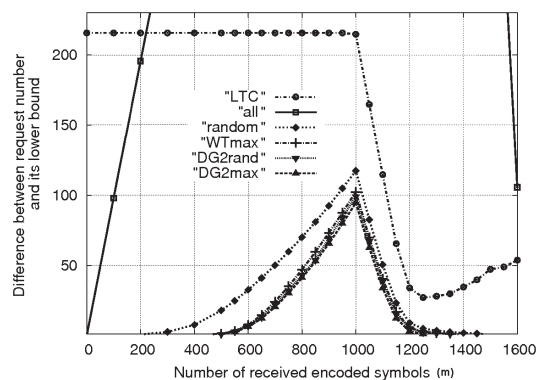


図 7 手法別平均要求回数と下界値 $l(m)$ との差 ($k = 1,000$)
Fig. 7 Difference between average request number and its lower bound ($k = 1,000$).

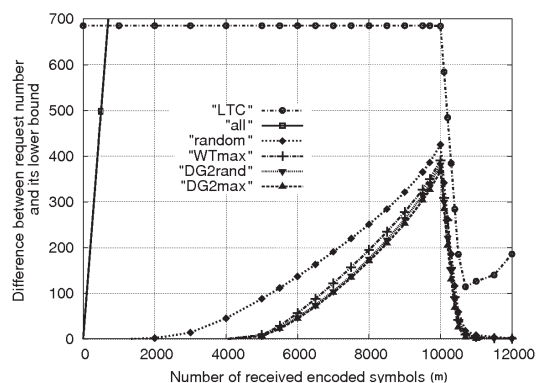


図 8 手法別平均要求回数と下界値 $l(m)$ との差 ($k = 10,000$)
Fig. 8 Difference between average request number and its lower bound ($k = 10,000$).

ンボル数が入力シンボルの 50% ~ 110%程度 ($m = 500 \sim 1,100$, 図 7; $5,000 \sim 11,000$, 図 8) の場合、*WTmax*手法、*DG2rand*手法、*DG2max*手法の性能が優れている。

5.3 計算量

各要求データ選択手法の計算量について議論する。本シミュレーション条件における、一回の要求シンボルの選択と復号にかかる総計算量の平均は、以下の順で多くなる。

$$all < LTC < random < DG2rand < WTmax < DG2max \quad (3)$$

*all*手法は復号を用いないため計算量は最も少なく、*LTC*手法は選択を行わないので次に少ない。*random*手法は、復号失敗時の未復号入力シンボル数が非常に多い傾向があるため (5.1)、次数 2 の符号化シンボルを検索する *DG2rand*手法より検索が少ない。*WTmax*手法は、全ての未復号入力シンボルの重みを調べる必要があるため *DG2rand*手法より検索が多い。*DG2max*手法は、全ての次数 2 の XOR 演算関係、全て未復号入力シンボルの

XOR 演算関係, もしくは全ての符号化シンボルの XOR 関係を調べる必要があるので検索が一番多い.

5.4 適切な選択手法の使い分け

5.1~5.3 の考察より, 配信未了者は受信符号化シンボル数に応じて適切な選択手法を用いると, ほぼ最適な要求回数をより少ない計算量で実現することが可能となる. 例えば, $k = 1,000$ において, 受信符号化シンボル数 $m \leq 300$, $m \geq 1,200$ の場合は *random* 手法を使用し, $300 < m < 1,200$ の場合は *DG2rand* 手法を使用する.

6. む す び

本稿では, LT 符号を用いた蓄積型マルチキャストコンテンツ配信において, マルチキャスト配信時間が限定されるシナリオに着目し, 全受信者の配信完了を保証するためのユニキャストベースの保証方式における要求データ選択手法について検討した. 本シナリオでは, マルチキャスト配信後に配信未了者が発生した場合, 配信未了者は配信が完了するまで送信者に差分データを追加転送して貰う必要がある. そこで我々は, 配信未了者が自ら有効な差分データを選択・要求し, 要求データ受信後に復号を行うこと (復号方針) で, 少ない追加転送回数 (要求回数) で配信が完了すると考え, 配信未了者向けの 3 種類の要求データ選択手法を提案した. 提案手法は, 上記復号方針と (1) 入力シンボルの選択, (2) 重みの大きな入力シンボルの選択, (3) 次数 2 の符号化シンボルに接続している入力シンボルの選択, の 3 つの選択方針に基づいている.

さらに, 計算機シミュレーションを行い, 提案手法の有効性を明らかにした. 例えば, 1,000 パケットのコンテンツ配信において, マルチキャストで 900 パケット程度しか受信できなかった配信未了者が提案手法を用いた保証方式を使用することで, 平均要求回数を比較手法 (*random* 手法) に比べて, 10~15%以上削減することができる. また, 各手法を比較することで (提案手法以外の手法含む), 復号方針と 3 つの選択方針の妥当性と複数の方針を用いることの有効性を確認した. また, マルチキャストの受信符号化シンボル数が非常に多い場合や少ない場合は, 3 種類の提案手法は理論限界に近い性能を達成することを示した. したがって, 適切な要求データ選択手法を用いることで, このような受信符号化シンボル数を有する配信未了者は, ほぼ最適な要求回数をより少ない計算量で実現することが可能となる.

文 献

[1] S. Floyd, V. Jacobson, S. McCanne, C. Liu, and L. Zhang, "A Reliable Multicast Framework for Lightweight Sessions and Application Level Framing," IEEE/ACM Trans. Netw., vol.5, no.6, pp.784-803, Dec. 1997.

[2] J. Nonnenmacher, E. Biersach, and D. Towsley, "Parity-based loss recovery for reliable multicast transmission," Proc. of ACM SIGCOMM'97, pp.289-299, Cannes, France, Sept. 1997.

[3] "Forward error correction (FEC) building block," IETF RFC3542, Dec. 2002.

[4] "The use of forward error correction in reliable multicast," IETF RFC3543, Dec. 2002.

[5] R. G. Gallager, "Low-density parity check codes," IRE Trans. Theory, IT-8, pp.21-28, 1962.

[6] M. Luby, "LT Codes," 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002.

[7] A. Shokrollahi, "Raptor codes," Digital Fountain, Inc., Tech. Rep. DF2003-06-001, June 2003.

[8] P. Maymounkov, "Online codes," NYU Technical Report TR2002-833, Nov. 2002.

[9] C. sasaki, T. Hasegawa, O. Kobayashi, S. Ano, T. Hasegawa, "On Unicast based Recovery for Multicast Content Distribution considering XOR-FEC," In Proc. APCC2005, pp.1043-1047, Oct. 2005.

[10] C. E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J. vol. 27, pp.379-423(Part I), pp.623-656(Part II), 1948.

[11] J. Ha and S. McLaughlin, "Optimal puncturing of low-density paritycheck codes," In Proc. IEEE ICC, 2003.

[12] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-dimensionally equidistributed uniform pseudorandom number generator," ACM Trans. on Modeling and Computer Simulation, vol.8, pp.3-30, Jan. 1998.

[13] J. Takahashi, H. Tode, K. Murakami, "An Efficient Multicast Distribution Scheme Combining Erasure Codes and Direct Requests," IEICE Trans. Commun., vol.E88-B, no.4, April 2005.