(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4787167号
(P4787167)
(45)発行日　平成23年10月5日(2011.10.5)　(24)登録日　平成23年7月22日(2011.7.22)

(51)Int.Cl.　F I
H03M 13/15 (2006.01)　H03M 13/15

請求項の数 8　(全 21 頁)

(21)出願番号	特願2006-541501 (P2006-541501)	(73)特許権者	501114844
(86)(22)出願日	平成16年12月1日 (2004.12.1)		デジタル　ファウンテン，　インコーポレイテッド
(65)公表番号	特表2007-513549 (P2007-513549A)		アメリカ合衆国、カリフォルニア州　９２１２１，サン・ディエゴ、モアハウス・ドライブ　５７７５
(43)公表日	平成19年5月24日 (2007.5.24)	(74)代理人	100108855
(86)国際出願番号	PCT/US2004/040271		弁理士　蔵田　昌俊
(87)国際公開番号	WO2005/055016	(74)代理人	100091351
(87)国際公開日	平成17年6月16日 (2005.6.16)		弁理士　河野　哲
審査請求日	平成19年11月22日 (2007.11.22)	(74)代理人	100088683
(31)優先権主張番号	60/526,218		弁理士　中村　誠
(32)優先日	平成15年12月1日 (2003.12.1)	(74)代理人	100109830
(33)優先権主張国	米国(US)		弁理士　福原　淑弘
(31)優先権主張番号	60/526,452		
(32)優先日	平成15年12月2日 (2003.12.2)		
(33)優先権主張国	米国(US)		

最終頁に続く

(54)【発明の名称】サブシンボル・ベース符号を使用する消去からのデータの保護

(57)【特許請求の範囲】
【請求項１】
ｓ個の入力シンボルを含むデータをｔ個の出力シンボルに符号化する方法であって、ｓは１より大きい整数であり、ｔは１より大きい整数であり、
前記ｓ個の入力シンボルのそれぞれを同一サイズのｍ個のサブシンボルに分割することと、
ｔ行ｓ列の基礎行列を取得することであって、前記基礎行列の項目は、有限体ＧＦ（２ｍ）の要素であり、前記基礎行列は、リードソロモン基礎行列ではない行列を形成する、ことと、
前記基礎行列の項目ごとに有限体の正則表現をＧＦ（２）モジュールとして置換し、これによってｔ＊ｍ行ｓ＊ｍ列の２進行列を生成することによって、前記基礎行列から拡大２進行列を生成することと、
ｔ＊ｍ個の出力サブシンボルを形成するために前記ｓ＊ｍ個の入力サブシンボルに対して前記拡大２進行列を作用させることと、
ｍ個の出力サブシンボルのグループを出力シンボルにグループ化しｔ個の出力シンボルを形成することであって、該ｔ個の出力シンボルは受信機が前記ｔ個の出力シンボルの一部または全部から前記ｓ個の入力シンボルを再生成するように通信チャネルを介して転送され、該通信チャネルにおけるロスの単位は１出力シンボル以上である、ことと、
を含むことを特徴とする方法。
【請求項２】

前記通信チャネルを介して前記ｔ個の出力シンボルを転送することをさらに含むことを特徴とする請求項１に記載の方法。

【請求項３】

前記ｓ個の入力シンボルのための可能な出力シンボルの個数は、前記ｓ個の入力シンボルを再生成するのに必要な出力シンボルの個数と独立であることを特徴とする請求項１に記載の方法。

【請求項４】

少なくとも１つの出力サブシンボルは、２以上の入力サブシンボル、かつ、前記ｓ＊ｍ個の入力サブシンボルの全部より少ない入力サブシンボル、かつ、少なくとも１つの入力シンボル内のｍ個のサブシンボルの全部より少ないサブシンボルから生成されることを特徴とする請求項１に記載の方法。

【請求項５】

前記基礎行列は、ランダム符号を表すことを特徴とする請求項１に記載の方法。

【請求項６】

前記基礎行列は、素体上における少なくとも次数２の拡大体である有限体上の連鎖反応符号を表すことを特徴とする請求項１に記載の方法。

【請求項７】

前記連鎖反応符号は、ランダム符号であることを特徴とする請求項６に記載の方法。

【請求項８】

前記基礎行列は、０より大きいジーナスの曲線に基づく代数幾何符号を表すことを特徴とする請求項１に記載の方法。

【発明の詳細な説明】

【技術分野】

【０００１】

本願は、その全体が本明細書に示されているかのように参照によって本明細書に組み込まれている、次の同時係属の米国仮出願の優先権を主張するものである。２００３年１２月１日出願の米国仮出願第６０／５２６，２１８号、「Protection of Data From Erasures Using Interleaved Transformations and Codes From Algebraic Geometry」（整理番号１９１８６－００５４００ＵＳ）、および、２００３年１２月２日出願の米国仮出願第６０／５２６，４５２号、「Protection of Data From Erasures Using Interleaved Transformations and Codes From Algebraic Geometry」。

【０００２】

本願は、その全体が本明細書に示されているかのように参照によって本明細書に組み込まれている、本願の譲受人に譲渡された次の米国特許および米国特許出願も参照する。ルビーに発行された米国特許第６３０７４８７号、「Information Additive Code Generator and Decoder for Communication Systems」（以下では「ルビーＩ」と呼称する）、およびショクロラヒらの米国特許第＿＿号（２００１年１２月２１日出願の米国特許出願第１０／０３２，１５６号）「Multi-Stage Code Generator and Decoder for Communication Systems」（以下では「ショクロラヒＩ」と呼称する）。

【背景技術】

【０００３】

障害のあるネットワークを介するデータの転送は、多くの研究の対象であった。インターネットなどの多くのコンピュータのネットワークまたは他のパケットベース・ネットワークでは、データは、まずパケットに副分割され、次にパケットを独立にネットワークを介して転送先に経路指定されることによって転送される。そのようなネットワークで、しばしば、パケット消失が見込まれる。パケットは、転送の物理層での誤り、機器にパケットを捨てさせるルータまたは他のネットワーク点でのオーバーフローなどに起因して失われる場合がある。データが完全に受信されることを保証するために、しばしば、そのような消失からデータを保護する手法が使用される。一般に、消失の単位は、パケットが正しく受信されるかパケットが消失したと考えられるのいずれかであるという点で、パケット

であり、パケット全体の消失を扱うためにステップが行われる。したがって、パケットのビットが受信されたが、パケットが完全には正しく受信されていない場合に、パケット全体が消失したとみなされる。消失は、パケット全体が欠ける形になることがあり、あるいは、パケットに誤りがあり、信頼できないビットが作られると判定されるという形すなわち、消去および誤りの形になることがある。

【０００４】

近年、データが転送中に失われる見込みがある時にデータを保護するために、２タイプの符号すなわち、連鎖反応符号およびマルチステージ連鎖反応符号が提案された。ｋ個のシンボルを有する所与のコンテンツについて、これらの符号は、元のｋ個のシンボルの回復が、その累積個数がおおむねｋと等しい別個の出力シンボルの任意のセットの受信により可能な、出力シンボルの事実上無限のストリームを生成する。そうでないと示されない限り、本明細書で使用される「連鎖反応符号」への言及は、ルビーＩおよび／または他所に記載のものなどの連鎖反応符号に適用することができ、ショクロラヒＩに記載のものなどのマルチステージ連鎖反応符号にも適用できることを理解されたい。

【０００５】

連鎖反応符号を用いると、ｋ個の入力シンボルの所与のセットについて可能な出力シンボルの個数が「事実上無限」と言われるのは、ほとんどすべての場合に、可能な出力シンボルの個数を、実際に生成される出力シンボルの個数に対して非常に多くすることができ、あるいは、入力シンボル回復に使用される可能な出力シンボルの個数が、可能なシンボルの個数よりはるかに少ないからである。たとえば、１００００ビットに関する入力シンボル符号と、通常の期待される転送が、サイズにおいて１０ギガビットまでのファイルまたはストリームである場合に、符号器は、ｋ＝１０００００００個のシンボルの入力を処理するように設計されなければならない。そのような符号器を、繰り返す必要なしに2^{32}（４０億）個までの出力シンボルを生成できるように構成することができる。それが十分でない場合に、符号器を、繰り返す必要なしにより多くの出力シンボルを生成できるように構成することができる。もちろん、すべての物理的に実現可能なシステムは有限なので、符号器は、最終的に、繰り返す状態に達するが、必ず、すべての期待される転送および誤り率について、繰り返しなしの出力シンボルの個数が事実上無限になるように、その状態を設計することができる。

【０００６】

本明細書では、パケットは、１つのシンボルまたは複数のシンボルを担持することができる。必須ではないが、１つの入力シンボルについて符号化されるビットの個数および１つの出力シンボルについて符号化されるビット数を、同一にすることができる。

【０００７】

いくつかの実施形態で、これらの符号は、入力シンボルに対して排他的論理和（以下ではＸＯＲと表記する）を実行することによってデータを符号化し、受信されたシンボルに対してＸＯＲを実行することによって復号するが、他の演算を、同様にまたはその代わりに使用することができる。ＸＯＲは、高速に逆演算可能であるため有用な演算である。他の演算も、これらの利益を提供することができる。

【０００８】

これらの符号は、消失率が送信機または受信機に未知の、障害のあるネットワーク上で１以上の送信機から１以上の受信機へデータを配信するという問題を解決する。この理由の１つは、入力シンボルの個数に対して可能な多数の出力シンボルを用いて、受信機が、圧倒的な可能性で、受信機間の調整なしに、別の受信機によって送信されるパケットを重複しないことである。この特性を、受信機が「情報加法的（ｉｎｆｏｒｍａｔｉｏｎ　ａｄｄｉｔｉｖｅ）」であると称する。

【０００９】

いくつかの場合に、所与のコンテンツから事実上無限の個数の出力シンボルを作ることが不必要または望ましくない場合がある。たとえば、受信機が時間的に制約されている場合に、所与の時間間隔の後に追加のシンボルが到着するのを待つという贅沢を有すること

はできない。これは、たとえば、ライブ映像が１つまたは複数の受信機に送信される時にあてはまる。ライブ転送の性質のゆえに、受信機の供給を送信機の要求と同期化しなければならず、無期限に中断することができないので、十分な符号化されたデータが受信機に到着するまで必ず待つことが、不可能である場合がある。その場合に、消失の可能性がある場合に、送信機は、コンテンツに固定された追加の量の冗長シンボルを追加し、そのコンテンツを冗長シンボルと一緒に転送することができる。コンテンツの転送中の消失の量が、冗長シンボルの個数より多くない場合には、受信機での消失データの回復の見込みがある。

【００１０】

この問題を、連鎖反応符号を用いて解決することもできる。その場合に、符号器は、事実上無限のストリームではなく、固定された量の符号化されたデータだけを生成する。しかし、いくつかの場合に、異なる解決策が好ましい場合がある。たとえば、連鎖反応符号の復号処理の確率的性質に起因して、これらの処理は、非常に小さいコンテンツサイズについてある追加のオーバーヘッドをこうむる場合がある。

【００１１】

リードソロモン符号（「ＲＳ符号」）は、コーダ出力と復号器入力の間で消去をうけるデータの転送または保管を扱うのに使用されてきた種類の符号である。本開示全体を通じて、符号化が、転送に制限されるのではなく、時間、場所などにおいて、符号化されたデータがチャネルを通過する際に消去および／または誤りを示す場合があるチャネルによって復号器から分離された符号器での元のデータを表す符号化であることを理解されたい。ＲＳ符号は、多数の研究者によって、多数の条件、データ、およびチャネルについて徹底的に研究され、ある特性を有することが知られている。

【００１２】

そのような条件の１つが、「最適性条件」として記述されるものである。ＲＳ符号は、２進数の体ではなくより大きいガロア体に作用する。ＲＳ符号の基本的特性の１つは、ＲＳ符号が、ｋ個のシンボルがＲＳ符号を用いて符号化され、格納または転送のためにｎ＜ｋ個のシンボルが作られる時に、元のｋ個のシンボルを、符号化されたｎ個のシンボルのｋ個の別個の受信されたシンボルのすべての可能な組合せから確実に回復できるという最適性条件を満足することである。元のｋ個のシンボルを、ｋ個未満の別個の受信されたシンボルから回復することはできないので、受信されるシンボルの個数は、したがって、「最適」と考えられる。

【００１３】

この最適性は、符号化に必要な動作の数が、より大きい、より長い符号（すなわち、より大きいガロア体）についてより多くなるという点で、代償を伴う。ＲＳ符号を用いると、最大ブロック長ｎは、前もって決定され、ここで、ブロック長は、元のｋ個の入力シンボルから生成される出力シンボルの個数である。ｎ－ｋ個を超える出力シンボルが失われた場合に、元のｋ個の入力シンボルを回復できないことに留意されたい。ブロック長ｎを、期待される条件を扱うために任意に長くすることはできない。というのは、計算が、より大きいブロック長についてよりむずかしくなり、非常に大きいブロック長について非実用的であるからである。

【００１４】

ガロア体ＧＦ（2^A）上で定義され、ブロック長ｎおよび次元ｋを有するリードソロモン符号について、１つの出力シンボルを生成するためのシンボルのＸＯＲの数は、平均してｋ＊（ｎ－ｋ）＊Ａ／（２＊ｎ）と等しい。そのようなリードソロモン符号を使用すると、ｋ個の入力シンボルが、合計ｎ個の出力シンボルを作るのに使用され、ここで、通常、ｋ個の入力シンボルが、ｎ個の出力シンボルに含まれ、ｎはｋより大きい。対照的に、連鎖反応符号を使用する時には、１つの出力シンボルを作るためのシンボルのＸＯＲの平均回数は、ｋまたは生成される出力シンボルの個数と独立の定数と等しい。類似する結果が、復号器にもあてはまる。

【００１５】

リードソロモン符号の長さｎは、2^A+1を超えることができない。この後者の条件が、Ａがしばしば２のべきになるように選択されるという事実と共に、符号化処理および復号処理を時にかなり低速にする場合がある。たとえば、元のコンテンツが、サイズにおいて３２ＫＢであり（ここで、１ＫＢ＝１０２４バイトである）、各パケットが、１ＫＢの入力データの符号化のために符号化し、合計４８個のパケットが送信されると仮定する。この例では、コンテンツを、３２個の１ＫＢチャンクに区分することができ（それぞれが送信される１つのパケットに割り振られる）、次に、各チャンクを、さらに、Ｘ個の入力シンボルに副分割することができる。次に、リードソロモン・符号化処理を、並列にＸ回適用し、各回に、各チャンクから１つの入力シンボルを操作することができ（各チャンクの最初の入力シンボルのすべてを操作し、次に各チャンクの２番目の入力シンボルなど）、これは、各操作が、３２個の入力シンボルを考慮に入れることを意味する。これによって、Ｘ個の位置のそれぞれについて１６個の追加の出力シンボルが生成され、Ｘ個の出力シンボルの各グループが、一緒に置かれて、それぞれが長さ１ＫＢの、送信される１６個の追加パケットが作られると仮定する。この例では、２のべき乗である最小の受入可能なＡは、Ａ＝４では$2^A+1=17$になり、４８より小さいので、Ａ＝８になる。この場合のリードソロモン符号は、体ＧＦ（２５６）で動作し、したがって、各シンボルが、１バイト長であり、Ｘ＝１０２４である。この例によって示されるように、これらの符号は、最適性条件を満足することはできるが、かなりの計算を必要とし、可能な符号の長さに対する制約を有する。

【００１６】

符号化のいくつかの概念が紹介に耐える。転送粒度は、１単位として転送され、受信されるオブジェクトのサイズを指す。たとえば、パケット・ネットワークは、パケットでデータを送信し、受信する。あるパケットのビットの一部だけが消去されるか壊された場合であっても、パケット全体が破棄され、機構（前方誤り訂正、再送信要求など）がアクティブ化されて、パケット全体を回復する。したがって、そのようなオブジェクトは、誤りなしで受信されるか、全体を消去されるのいずれかになる。いくつかの応用例で、オブジェクト・サイズを、転送パケットのサイズとすることができ、あるいは、それより小さくすることができる。転送パケットの間で消失の相関の見込みがある場合に、転送粒度を、パケット・サイズより大きくすることができる。他の応用例で、転送粒度を、パケット・サイズより小さくすることができる。

【００１７】

計算粒度は、符号器および／または復号器で操作されるオブジェクトのサイズを指す。したがって、符号器の基本動作が１２８バイト単位のＸＯＲである場合に、それが計算粒度である。１２８バイト・サブシンボルに副分割される１０２４バイトを含むシンボル（たとえばパケットとすることができる）が、８つのサブシンボルに分割されるシンボルであり（サブシンボルのすべてが同一サイズである必要がない場合があるが、そうである場合には、この方が単純である）、ＸＯＲは、これらのサブシンボルに対して実行される。したがって、計算粒度は、１２８バイトである。

【００１８】

リードソロモン符号の最適性の理由の１つが、その転送粒度と計算粒度の間の関係にある。１つの例が、このポイントを示す。

【００１９】

所与のファイルを符号化し、符号化された情報を、それぞれ１０２４バイトのサイズのパケットでチャネルを介して転送するのに使用される、体ＧＦ（２５６）上のリードソロモン符号を検討されたい。この場合の計算粒度は、１２８バイト（１０２４バイトを８で割る）と等しいものとすることができるが、転送粒度は、１０２４バイトと等しい。この場合に、ビットのシーケンスのＸＯＲなどの基本演算が、１２８バイトの単位全体に対して実行される。

【００２０】

通常、符号化および復号の効率は、計算粒度に伴って変化する。効率は、多数の形で測

定することができるが、これを測定する１つの形が、データの単位を符号化または復号する動作の平均回数によるものである。しばしば、符号化および復号は、より微細な計算粒度についてより非効率的であり、より粗な計算粒度についてより効率的である。しかし、より微細な計算粒度を用いる符号は、より良い受信オーバーヘッドを提供できる、すなわち、符号器に供給されるデータを表すシンボルの個数に対する、正しい復号を保証するために受信される必要があるシンボルの個数の追加を非常に小さく保つことができる。その結果、所与の符号の符号化効率と転送オーバーヘッドの間にトレード・オフがある。

【００２１】

リードソロモン符号は、この符号化トレード・オフの一方の端にある。というのは、計算粒度が、消去が存在する場合にデータの最適回復が保証されるのに十分に小さいからである（符号化されたのと同量のデータの受信時に。他方の端で、２進アルファベット上で定義された符号（パケット・ネットワーク上での転送に使用されるものなど）は、転送粒度と同じ大きさの計算粒度を有するが、完全な復号を保証するのに必要な受信オーバーヘッドにおいて非効率的である場合があるからである。

【００２２】

上で述べたように、リードソロモン符号は、最大誤り率が前もって判定されることを必要とする、すなわち、ｋ個のシンボルがｎ個のＲＳシンボルに符号化される場合に、（ｎ－ｋ）／ｎを超える誤り率は、復号器に、転送されたデータの回復を失敗させる。したがって、転送されるデータの不成功の回復の最終的な確率によって測定される転送システムでは、リードソロモン符号は、その最適性にもかかわらず、正の失敗確率を示す。これは、受信機によって受信されるデータの量が、転送されたデータより本当に少ないことの正の確率があるからである。その結果、最終的に、符号化システムは、より低い効率の符号化を有し、低減の必要がある失敗確率を有する場合がある。

【特許文献１】米国仮出願第６０／５２６，２１８号
【特許文献２】米国仮出願第６０／５２６，４５２号
【特許文献３】米国特許第６３０７４８７号
【特許文献４】米国特許出願第１０／０３２，１５６号

【発明の開示】

【発明が解決しようとする課題】

【００２３】

したがって、必要なものは、計算労力およびオーバーヘッド効率を、特定の応用例、使用可能な処理能力、およびデータ・セットの必要に応じてトレード・オフすることができ、チャネルを介して送信されるデータを符号化し、復号する符号化のシステムおよび方法である。

【課題を解決するための手段】

【００２４】

本発明による通信システムの一実施形態で、符号化器は、出力シンボルサブシンボルを使用して、計算労力とオーバーヘッド効率のトレードオフをもたらすか制御して、たとえば、少量のオーバーヘッド効率を犠牲にして計算労力を大幅に低減する。符号化器は、入力ファイルまたは入力ストリームを含む順序付きの複数の入力シンボルを読み取り、出力サブシンボルを作る。順序付きの複数の入力シンボルは、それぞれ、入力アルファベットから選択され、生成された出力サブシンボルに、出力サブシンボル・アルファベットの中での選択が含まれる。出力サブシンボルは、入力シンボルのサブシンボルに適用される関数評価器を使用して生成される。いくつかの実施形態で、符号化器を、１回または複数回呼び出すことができ、各回に、１つの出力サブシンボルが作られる。次に、出力サブシンボルを出力シンボルに組み立て、その転送先に転送することができる。

【００２５】

本発明の諸態様による１つの符号化処理では、入力サブシンボルから出力サブシンボルを生成するのに使用される関数が、いくつかの入力サブシンボルのＸＯＲである。本発明の諸態様によるもう１つの符号化処理では、これらの関数が、ＧＦ（２）上の拡大体の正

則表現を使用して、ＧＦ（２）の拡大体上で定義された線形符号の生成行列またはパリティ検査行列の各項目を適当な２進行列に変換することによって、この符号から得られる。

【００２６】

本発明の諸態様による復号器では、受信機によって受信された出力サブシンボルが、入力シーケンス（ファイル、ストリームなど）の符号化に基づいてこれらの出力シンボルを生成した１つの送信機から転送された出力シンボルから得られる。出力シンボルが、転送中に失われる可能性があるので、復号器は、転送された出力シンボルの任意の部分だけを受信した時であっても正しく動作する。

【００２７】

本発明は、計算労力と転送効率のトレード・オフを制御できることなどの利益を提供する。たとえば、緩和された最適性条件を用いると、可能な転送労力のわずかな増加で、非常に低減された計算労力とすることができる。ある符号を用いると、追加出力が簡単に入手可能になり、その結果、最大最適性条件の下での復号に必要な個数のシンボルに対する比較的少数の追加シンボルだけの受信を用いて、復号失敗率を任意に減らすことができる。実施形態で、計算の単位（個々の符号化動作または復号動作の一部であるデータの間の境界）および消失の単位（１単位の境界内のいずれかのデータが入手不能である場合に、その単位境界内のすべてのデータが失われたとみなされる、データの間の境界）を有することによって、計算労力が減る。特定の実施形態で、消失の単位は、シンボルまたはパケットであり、計算の単位は、サブシンボルである。

【００２８】

本明細書に開示される発明の性質および利益のさらなる理解は、本明細書の残りの部分および添付図面を参照することによって実現することができる。

【発明を実施するための最良の形態】

【００２９】

本明細書で説明する実施例では、「サブシンボル・ベース・符号化」と表される符号化方式を説明するが、その前に、この説明で使用されるさまざまな用語の意味および範囲を説明する。

【００３０】

符号器は、ファイル、ストリーム、または他の入力データ転送元から入力データを受け取り、チャネルが及ぼす可能性があるデータに対する影響をチャネルの他端の復号器によって埋め合わせることができ、復号器が、どのような精度が必要であれその精度まで元のデータを再生成できるようにそのデータを符号化する、ソフトウェア処理、ハードウェア装置、組合せ、または類似物である。

【００３１】

サブシンボル・ベース・符号化を用いると、出力シンボルは、送信機によって、必要に応じて入力ファイルから生成される。各出力シンボルに、１つまたは複数のサブシンボルが含まれ、少なくとも１つの出力シンボルに、少なくとも２つのサブシンボルが含まれる。出力シンボル内の各サブシンボルは、符号器または復号器のソフトウェアおよび／またはハードウェアを使用して、入力ファイルを含むシンボルのサブシンボルに対して計算動作を実行することによって生成される。生成されたならば、出力シンボルを、パケットに置き、その転送先に転送することができ、各パケットに、１つまたは複数の出力シンボルが含まれる。

【００３２】

本明細書で使用する用語「ファイル」は、１以上の転送元に格納され、１単位として１以上の転送先に配信されるすべてのデータを指す。したがって、ファイル・サーバまたはコンピュータ記憶装置からの文書、イメージ、およびファイルのすべてが、配布できる「ファイル」の例である。ファイルは、既知のサイズである（ハード・ディスクに保管された１メガバイト・イメージなど）ものとすることができ、あるいは、未知のサイズである（ストリーミング転送元の出力からとられるファイルなど）ものとすることができる。いずれであれ、ファイルは、入力シンボルのシーケンスであり、各入力シンボルは、ファイ

ル内の位置および値を有する。

【００３３】

転送とは、ファイルを配布するために１つまたは複数の送信機から１つまたは複数の受信機へチャネルを介してデータを転送する処理である。１つの送信機が、完全なチャネルによって任意の個数の受信機に接続される場合に、受信されるデータは、すべてのデータが正しく受信されるので、入力ファイルの正確なコピーでありえる。本明細書では、チャネルが不完全であると仮定され、これは、ほとんどの実世界のチャネルにあてはまる。多数のチャネル不完全性のうちで、関心を持たれる２つの不完全性は、データ消去およびデータ不完備性（データ消去の特殊な事例として扱うことができる）である。データ消去は、チャネルがデータを失うか捨てる時に発生する。データ不完備性は、受信機が、データの一部が既にそれに渡された後になるまで受信を開始しない時、受信機が、転送が終了する前にデータの受信を停止する時、または、受信機が、データの受信を間欠的に停止し、再び開始する時に発生する。データ不完備性の例として、移動する衛星送信機は、入力ファイルを表すデータを送信しており、受信機が範囲内になる前に転送を開始する場合がある。受信機が範囲内になったならば、データを、衛星が範囲外になるまで受信することができ、この時点で、受信機は、その衛星パラボラアンテナを向けなおして（その間に、受信機はデータを受信してない）、範囲内に移動した別の衛星によって転送される同一入力ファイルに関するデータの受信を開始することができる。

【００３４】

この説明を読むことから明白になるように、データ不完備性は、データ消去の特殊な事例である。というのは、受信機は、受信機がつねに範囲内にあったが、チャネルが、受信機がデータ受信を開始した点までのすべてのデータを失ったかのように、データ不完全性を扱うことができるからである（受信機は、同一の問題を有する）。また、通信システム設計で周知のように、検出可能な誤りは、検出可能な誤りを有するすべてのデータ・ブロックまたはシンボルを単純に捨てることによって、消去と同等とすることができる。

【００３５】

一般に、転送は、送信機から受信機へ、送信機と受信機を接続するチャネルを介してデータを移動する操作である。チャネルは、リアルタイム・チャネルとすることができ、ここで、チャネルは、チャネルがデータを得た時に送信機から受信機にデータを移動し、あるいは、チャネルを、送信機から受信機にデータを送る際にデータの一部またはすべてを格納するストレージ・チャネル（ｓｔｏｒａｇｅ　ｃｈａｎｎｅｌ）とすることができる。後者の例が、ディスク・ストレージまたは他の記憶装置である。その例では、データを生成するプログラムまたは装置を、データを記憶装置に転送する送信機と考えることができる。受信機は、その記憶装置からデータを読み取るプログラムまたは装置である。送信機がデータを記憶装置に置くメカニズム、記憶装置自体、および受信機が記憶装置からデータを得るために使用するメカニズムが、集合的にチャネルを形成する。これらのメカニズムまたは記憶装置がデータを失う可能性がある場合に、これを、そのチャネルでのデータ消失として扱うことができる。

【００３６】

＜１．基本的実装＞

通常の実装では、サブシンボル・ベース・符号化を使用するファイルの転送に、入力ファイルからの入力シンボルの生成、形成、または抽出と、入力シンボルのそれぞれのサブシンボルの生成と、これらのサブシンボルの１つまたは複数の出力サブシンボルへの符号化と、出力サブシンボルからの出力シンボルの作成と、１つまたは複数の受信機へのチャネルを介する出力シンボルの転送とが含まれる。

【００３７】

サブシンボル・ベース・符号化を使用する入力ファイルのコピーの受信（および再構成）に、１つまたは複数のデータ・ストリームから出力シンボルのあるセットまたはサブセットを受信することと、受信された出力シンボルのそれぞれについてサブシンボルを生成することと、受信された出力サブシンボルの値から入力サブシンボルを復号することと、

復号された入力サブシンボルから入力シンボルを作成することと、入力シンボルから入力ファイルを再組立することとが含まれる。いくつかの実施形態で、たとえば復号された入力サブシンボルから直接に入力ファイルを再組立することができる場合に、最後のステップを破棄することができる。

【００３８】

本発明の諸態様を、これから図面を参照して説明する。

【００３９】

図１は、サブシンボル・ベース・符号化を使用する通信システム１００のブロック図である。通信システム１００では、入力ファイル１０１または入力ストリーム１０５が、サブシンボル生成器１１０に供給される。サブシンボル生成器１１０は、入力ファイルまたは入力ストリームから１つまたは複数の入力サブシンボル（ＩＳ（０，０）、ＩＳ（０，１）、ＩＳ（０，２）、・・・）のシーケンスを生成し、各入力シンボルは、１つの値および２つの位置（図１では括弧を付けた整数として表される）を有する。サブシンボル生成器１１０は、その入力の１つとして値ｍを使用し、この値ｍは、各入力シンボルまたは各出力シンボル内のサブシンボルの個数である。サブシンボル生成器の出力は、それぞれｍ個のグループに分割され、各グループの要素は、第２の括弧内の整数を用いて識別され、これは、０とｍ－１の間の整数である。

【００４０】

上で説明したように、各入力サブシンボルのサイズは、符号化システムの計算粒度であり、転送粒度は、計算粒度のｍ倍以上の任意の数とすることができる。本明細書で提供する例では、しばしば、説明を単純にするためにサブシンボルのサイズがすべて等しいと仮定するが、このサイズを変更することができ、正しい機能のために一定のサイズが必要ではないことを理解されたい。

【００４１】

入力サブシンボルの可能な値すなわち入力サブシンボルのアルファベットは、通常、2^M個のシンボルのアルファベットであり、その結果、各入力サブシンボルが、入力ファイルのＭビットについて符号化するようになる。Ｍの値は、一般に、通信システム１００の用途によって決定されるが、汎用システムには、サブシンボル生成器１１０のサブシンボル・サイズ入力を含めることができ、その結果、Ｍを、使用ごとに変更することができるようになる。サブシンボル生成器１１０の出力は、符号器１１５に供給される。

【００４２】

符号器１１５は、入力サブシンボル生成器１１０によって供給される入力サブシンボルから、値ＯＳ（ｉ，ｊ）を有する出力サブシンボルを生成する。各出力サブシンボルの値は、本明細書で出力サブシンボルの「関連する入力シンボル」または単にその「付随物」と称する、１つまたは複数の入力サブシンボルのある関数に基づいて生成される。関数（「値関数」）および付随物の選択は、下で詳細に説明する処理に従って行われる。必ずではないが、通常、Ｍは、入力サブシンボルおよび出力サブシンボルいついて同一である、すなわち、この両方が、同一のビット数について符号化する。

【００４３】

いくつかの実施形態で、入力サブシンボルの個数Ｋが、符号器によって、付随物を選択するのに使用される。入力がストリーミング・ファイルである場合など、Ｋが前もって既知でない場合には、Ｋを、推定値とすることができる。値Ｋは、符号器１１５によって、入力サブシンボルのストレージを割り振るのに使用することもできる。

【００４４】

符号器１１５は、出力サブシンボルを出力シンボル生成器１３５に供給する。出力シンボル生成器１３５には、各シンボル内のサブシンボルの個数ｍも供給される。出力シンボル生成器１３５は、図１でＯＳ（０）、ＯＳ（１）、・・・などとして示された出力を送信モジュール１４０に供給する。送信モジュール１４０は、チャネル１４５を介して受信モジュール１５０に出力シンボルを転送する。チャネル１４５は、消去チャネルであると仮定されるが、これは、通信システム１００の正しい動作の要件ではない。モジュール１

４０、１４５、および１５０は、送信モジュール１４０が、出力シンボルおよびそのキーに関するすべての必要なデータをチャネル１４５に転送するように適合され、かつ、受信モジュール１５０が、シンボルおよびそのキーに関する潜在的ないくつかのデータをチャネル１４５から受信するように適合されている限り、任意の適当なハードウェア構成要素、ソフトウェア構成要素、物理媒体、またはその任意の組合せとすることができる。Ｋの値は、付随物を決定するのに使用される場合に、チャネル１４５を介して送信することができ、あるいは、事前に、符号器１１５および復号器１５５の合意によってセットすることができる。

【００４５】

上で説明したように、チャネル１４５を、インターネットを介する経路またはテレビジョン送信機からテレビジョン受信機への放送リンクまたはある点から別の点への電話接続など、リアルタイム・チャネルとすることができ、あるいは、チャネル１４５を、ＣＤ－ＲＯＭ、ディスク・ドライブ、ウェブ・サイト、または類似物などのストレージ・チャネルとすることができる。チャネル１４５は、ある人がパーソナル・コンピュータからインターネット・サービス・プロバイダ（ＩＳＰ）へ電話回線を介して入力ファイルを転送し、入力ファイルがウェブ・サーバに保管され、その後、インターネットを介して受信機に転送される時に形成されるチャネルなど、リアルタイム・チャネルとストレージ・チャネルの組合せとすることさえできる。

【００４６】

チャネル１４５は、消去チャネルであると仮定されるので、通信システム１００は、受信モジュール１５０を出る出力シンボルと送信モジュール１４０に入る出力シンボルとの間の１対１対応を仮定しない。実際には、チャネル１４５にパケット・ネットワークが含まれる場合に、通信システム１００は、複数のパケットの相対的な順序が、チャネル１４５を介して移動する際に保存されると仮定することすらできない場合がある。

【００４７】

受信モジュール１５０は、受信したシンボルＲＳ（０）、ＲＳ（１）をサブシンボル生成器１６０に供給する。この生成器には、各受信された出力シンボルに含まれるサブシンボルの個数の値ｍも与えられる。この情報は、送信機と受信機の間の転送の前に共有することができ、転送の一部とすることができ、あるいは、受信機がそれを知らず、受信機が即座に復号する必要がない場合に後で供給することができる。前に述べたように、ｍの値は、すべての受信された出力シンボルについて同一でないものとすることができる。

【００４８】

サブシンボル生成器１６０は、ＲＳ（０，０）、ＲＳ（０，１）、・・・などと表された、復号器１５５への出力を生成する。各受信されたシンボルにｍ個のサブシンボルが含まれる場合に、サブシンボル生成器１６０の出力が、それぞれｍ個のグループに分割され、各グループは、各受信されたシンボル内のサブシンボルに対応する。第２の括弧内の整数は、受信されたシンボル内のサブシンボルの位置に対応し、第１の整数は、出力されるサブシンボルがそのサブシンボルである受信されたシンボルに対応する。この場合に、サブシンボル生成器の出力は、ＲＳ（０，０）、・・・、ＲＳ（０，ｍ－１）、ＲＳ（１，０）、・・・、ＲＳ（１，ｍ－１）などである。

【００４９】

復号器１５５は、サブシンボル生成器１６０によって供給された出力サブシンボルを使用して、入力サブシンボル（やはりＩＳ（０，０）、ＩＳ（０，１）、ＩＳ（０，２）、・・・）を回復する。復号器１５５は、回復された入力サブシンボルをシンボル生成器１６２に供給し、シンボル生成器１６２は、入力シンボルＩＳ（０）、ＩＳ（１）、・・・などを作る。これらの入力シンボルが、入力ファイル再組立器１６５に供給され、入力ファイル再組立器１６５は、入力ファイル１０１または入力ストリーム１０５の入力ファイル１７０を生成する。いくつかの応用例で、シンボル生成器１６２を迂回し、出力を入力ファイル再組立器１６５に直接に転送することができる。

【００５０】

＜２．基本的符号器＞
図２は、基本的な符号器の例示的なブロック図である。生成される出力シンボル２３０ごとに、Ｆ（ｉ，ｊ）によって表される関数評価器２２０が含まれる。図２の例では、ｍ＝４であり、２０１、・・・、２１２によって表される、合計１２個の入力サブシンボルがある。関数評価器２２０は、入力サブシンボルの関数を計算して、出力サブシンボル２３０を生成する。たとえば、図２に示された状況では、関数評価器は、関数Ｆ（ｉ，ｊ）を使用して、入力ＩＳ（０，０）、ＩＳ（０，３）、ＩＳ（１，１）、ＩＳ（１，３）、およびＩＳ（２，２）から出力ＯＳ（ｉ，ｊ）の値を計算する。
【００５１】
いくつかの実施形態で、各生成される出力サブシンボルは、異なる関連する関数を有し、これらの関数は、事前に生成されるか、擬似乱数的に生成される。他の実施形態で、関数評価器２２０は、生成される出力シンボルの多くについて同一とすることができ、関数に使用される入力値のセットにおいてのみ異なるものとすることができる。たとえば、単純なインターリーブ方式が使用される場合に、Ｆ（ｉ，ｊ）を、ｊのすべての値について同一とし、入力値のセットにおいてのみ異なるものとすることができる。具体的には、この場合に、関数Ｆ（ｉ，ｊ）は、入力としてサブシンボルＩＳ（０，ｊ）、ＩＳ（１，ｊ）、ＩＳ（２，ｊ）などのみを使用する。
【００５２】
図２に開示されているように、関数評価器２２０は、その入力の任意の関数とすることができる。いくつかの実施形態で、具体的には線形符号を有することが望ましい実施形態で、この関数を、その引数の線形関数、たとえば引数のＸＯＲになるように選択しなければならない。「インターリーブ変換（ｉｎｔｅｒｌｅａｖｅｄ　ｔｒａｎｓｆｏｒｍａｔｉｏｎ）」と称する、関数評価器２２０によって使用できる線形関数のそのような種類の１つを、これから説明する。
【００５３】
＜３．インターリーブ変換＞
本明細書で説明するいくつかの処理は、Ｂｌｏｅｍｅｒ、Ｋａｌｆａｎｅ、Ｋａｒｐ、Ｋａｒｐｉｎｓｋｉ、Ｌｕｂｙ、およびＺｕｃｋｅｒｍａｎにより「An XOR Based Erasure Resilient Coding Scheme」、International Computer Science Institute (ICSI) Technical Report TR-95-048で暗黙のうちに述べられた方法を利用する。
【００５４】
ｓ個のシンボルに編成された入力データをｔ個のシンボルに編成された出力データに変換し、各シンボル（入力データおよび出力データの）に等しいサイズのｍ個のサブシンボルが含まれ、変換がｔ行ｓ列の基礎行列を使用し、各基礎行列項目が有限体ＧＦ（2^m）の値である変換処理を検討されたい。
【００５５】
この変換処理は、基礎行列の各項目をｍ行ｍ列の２進行列に変換することによって開始される。この変換では、ＧＦ（２）モジュールとして有限体の正則表現を利用するが、これは、有限の代数および符号化の理論の当業者に周知の概念であり、したがって、本明細書ではさらに詳細には説明しない。この変換を元の基礎行列のすべての項目に適用することによって、ｔ＊ｍ行ｓ＊ｍ列の新しい２進行列が作られる。この変換処理は、この新しい行列を入力データのｓ＊ｍ個のサブシンボルに適用して、新しい２進行列の行ごとに１つの、ｔ＊ｍ個の出力サブシンボルに達する。新しい２進行列の各行は、１つのサブシンボルに対応し、この変換処理は、その行および列に１がある各入力サブシンボルのＸＯＲをとることによって、所与の出力サブシンボルを決定する。この形で作成された最終的なｔ＊ｍ個のサブシンボルが、ｍ個のサブシンボルのグループにグループ化されて、それぞれ、ｔ個のシンボルが作られる。
【００５６】
この変換処理が、サブシンボルのＸＯＲだけを実行する必要があることに留意されたい。この変換処理が実行するＸＯＲの回数は、元の行列に依存するが、この回数は、平均し

てｓ＊ｔ＊ｍ／２と等しい。
【００５７】
上で説明した変換処理の例として、図３Ａに示された、体ＧＦ（４）＝｛０，１，α，α^2｝上のｓ＝５×ｔ＝２基礎行列を検討されたい。ＧＦ（４）について、ｍ＝２である。基礎｛１，α｝に関してＧＦ（２）上のＧＦ（４）の正則表現を使用すると、図３Ａの基礎行列は、図３Ｂに示された新しい２進行列に変換され、この新しい２進行列は、ｓ＊ｍ＝１０列×ｔ＊ｍ＝４行の行列である。
【００５８】
この変換行列を使用すると、それぞれが２つのサブシンボルを含む５つのシンボルを含む元のデータは、次のように変換される。最初の４つの出力サブシンボルのうちの最初のサブシンボルは、入力サブシンボル３、６、７、９、および１０が、図３Ｂの行列の第１行の「１」の位置なので、これらのサブシンボルのＸＯＲとして計算される。「１」が、２進値の１つの状態の任意の表示であり、したがって、それについて入力シンボルが使用されるラベルと考えられなければならないことに留意されたい。
【００５９】
第２の出力サブシンボルは、入力サブシンボル４、５、６、８、および９のＸＯＲとして計算される。第３の出力サブシンボルは、入力サブシンボル１、５、８および１０のＸＯＲとして計算される。最後に、最後（第４）の出力サブシンボルが、入力サブシンボル２、６、７、８、９、および１０のＸＯＲとして計算される。この場合に実行されるサブシンボルのＸＯＲの総数は、２０である。
【００６０】
この特定の例における関数Ｆ（ｉ，ｊ）は、次のように与えられる。
【００６１】
Ｆ（０，０）＝ＩＳ（１，０）＋ＩＳ（２，１）＋ＩＳ（３，０）＋ＩＳ（４，０）＋ＩＳ（４，１）
Ｆ（０，１）＝ＩＳ（１，１）＋ＩＳ（２，０）＋ＩＳ（２，１）＋ＩＳ（３，１）＋ＩＳ（４，０）
Ｆ（１，０）＝ＩＳ（０，０）＋ＩＳ（２，０）＋ＩＳ（３，１）＋ＩＳ（４，１）
Ｆ（１，１）＝ＩＳ（０，１）＋ＩＳ（２，１）＋ＩＳ（３，０）＋ＩＳ（３，１）＋ＩＳ（４，０）＋ＩＳ（４，１）
ここで、記号「＋」はＸＯＲ演算子を表す。
【００６２】
インターリーブ変換は、ＧＦ（２）の拡大体上の生成行列およびパリティ検査行列によって定義される符号を使用する符号器および／または復号器の実施形態の一部として使用することができる。
【００６３】
＜４．基本的出力シンボル生成器＞
図４は、出力シンボル生成器１３５のブロック図である。この図には、ｍ＝４および３つの出力シンボルの事例が例示されている。この例では、出力シンボル生成器１３５が、ＯＳ（ｉ，０）、ＯＳ（ｉ，１）、ＯＳ（ｉ，２）、・・・、ＯＳ（ｉ，ｍ－１）（この図では４０１、・・・、４１２として参照される）と表された４つの出力シンボルのグループを、出力シンボルＯＳ（ｉ）（この図では４２０、４３０、および４４０として参照される）にパックする。すべての出力シンボルに関する同一の値ｍの選択は、説明を単純にするために行われたものにすぎない。この値は、出力シンボル生成器が、生成される出力シンボルのｍの値の表示を有する限り、異なる出力シンボルについて異なるものとすることができる。
【００６４】
＜５．基本的サブシンボル生成器＞
図５は、基本的なサブシンボル生成器１６０のブロック図である。この図は、ｍ＝４およびおよび３つの受信された信号ＲＳ（０）、ＲＳ（１）、およびＲＳ（２）の事例が例

示されている。サブシンボル生成器１６０の動作は、図４で与えた出力シンボル生成器１３５の動作の逆に対応する。

【００６５】

図５に与えられた例では、サブシンボル生成器１３５が、受信されたシンボルごとに４つのサブシンボル（この図では５０１、・・・、５１２として参照されるサブシンボル）を作成する。サブシンボルＲＳ（ｉ，０）、・・・、ＲＳ（ｉ，３）は、受信されたシンボルＲＳ（ｉ）に対応する（この図では５２０、５３０、および５４０として参照される）。すべての受信されたシンボルに関する同一の値ｍの選択は、説明を単純にするために行われたものである。この値は、サブシンボル生成器が、すべての受信されたシンボルについてそのシンボルのｍの値の表示を供給される限り、異なる受信されたシンボルについて異なるものとすることができる。そのような表示は、送信機から帯域外情報を介して、または送信機および受信機によって共有される事前に決定されるアルゴリズムを介して、サブシンボル生成器に供給することができる。

【００６６】

＜６．ＧＦ（２）の拡大体に対するインターリーブ変換および符号を使用するサブシンボル・ベース・符号化＞

サブシンボル・ベース・符号化を、上で説明したように、本明細書で説明するインターリーブ変換と一緒に使用して、符号が計算および消去の粒度および労力に関するトレード・オフを示すように、消去の見込みがあるパケット・ネットワークでの転送用の符号を設計することができる。

【００６７】

一実施形態で、符号器および復号器が、拡大体ＧＦ（２^{m}）上で定義された符号を使用するように構成される。この符号は、生成行列、パリティ検査行列、または類似する結果に達するある抽象符号化処理もしくはルールのセットによって定義することができる。提示を単純にするために、本明細書の例では、生成行列を使用して、符号化を説明するが、他の手法を、同一の、類似する、または異なる結果のために使用できることを理解されたい。

【００６８】

生成行列が、ｎ行ｋ列を有すると仮定する。符号が組織的符号（ｓｙｓｔｅｍａｔｉｃ　ｃｏｄｅ）であると仮定するが、その代わりに、符号を非組織的符号とすることができることを理解されたい。

【００６９】

組織的符号を用いると、最初のｋ列からなる部分行列が、単位行列になる。残りのｒ＝ｎ－ｋ列からなる部分行列を、本明細書ではＣと称する。この行列Ｃは、ｒ行ｋ列を有する。符号化されるデータが、シンボル（またはパケット）ｋ個の長さであると仮定する。符号化処理は、上のインターリーブ変換処理を行列Ｃおよび符号化されるデータに適用する処理である。

【００７０】

前の符号化方法と比較したこの符号化方法の利益の１つは、転送方式のオーバーヘッド特性が、有限体ＧＦ（２^{m}）上の元の符号の構造によって左右されるが、計算が体ＧＦ（２）上で実行されることである。本明細書で説明するものなどの符号パラメータの賢明な選択のために、または本明細書で説明する選択方法によれば、上で説明したトレード・オフの優秀なバランスを提供する符号化構造を得ることが可能である。

【００７１】

復号処理は、下で説明するように、サブシンボルのゆえにより複雑であり、通常は複数のステップで実行する。特筆すべきことに、復号計算労力を、それでも、以前の方法が同一の誤り回復結果を得るのに使用される場合より少なくすることができる。

【００７２】

例の復号処理では、符号化されたデータ・パケットのそれぞれが、関連する位置を有し、これらの位置が、１からｎまでの整数によって表現可能であると仮定する。最初のｋ個

の位置は、組織的位置（ｓｙｓｔｅｍａｔｉｃ　ｐｏｓｉｔｉｏｎ）と呼ばれ、これらの位置の、転送の前の符号化されたデータは、符号化されるデータと同一である。残りの位置のデータ（またはパケット）を、冗長パケットと呼ぶ。前と同様に、最初のｋ行が単位行列を形成し、残りのｒ行が行列Ｃを形成する生成行列が与えられると仮定する。

【００７３】

復号処理の一実施形態で、ステップは次の通りである。

【００７４】

１）消去された組織的パケットの位置q_1、q_2、・・・、q_eを記録し、保管するが、ここで、ｅは、そのような消去されたパケットの個数である。そのようなパケットが存在しない場合には、復号の成功を宣言し、終了する。

【００７５】

２）カウンタｌ＝０をセットする。

【００７６】

３）復号が成功していない（まだ元のパケットのすべてが回復されてはいない）間は、下のサブステップ（ａ）から（ｅ）を実行する。

【００７７】

（ａ）ｅ＋ｌ個の消去されていない冗長パケットを見つける。ｅ＋ｌ個未満の消去されていない冗長パケットが使用可能である場合には、復号誤りを宣言する。そうでない場合には、ｅ＋ｌ個の消去されていない冗長パケットの位置を、p_1、p_2、・・・、p_{e+l}によって表す。

【００７８】

（ｂ）位置p_1、p_2、・・・、p_{e+l}に対応する行および位置q_1、q_2、・・・、q_eに対応する列を含む、生成行列の部分行列を形成する。これを行列Ｄと呼ぶ。この行列が、Ｃの部分行列であることに留意されたい。

【００７９】

（ｃ）たとえばガウス消去または明示的判定または他の方法によって、Ｄの可逆ｅ×ｅ部分行列を見つける。そのような部分行列が存在しない場合には、カウンタｌを１つ増分し、ステップ３に進む

（ｄ）そのような可逆部分行列が存在する場合には、その行r_1、・・・、r_eを記録し、基礎体上でのその逆行列を計算し、これを逆行列Ｂと呼ぶ。

【００８０】

（ｅ）インターリーブ変換処理を行列Ｂおよび位置r_1、・・・、r_eに冗長パケットを含むデータに適用して、位置q_1、q_2、・・・、q_eの消去された組織的パケットを得る。復号成功を宣言し、停止する。

【００８１】

次は、値の１特定のセットを使用する、この復号処理の詳細な例である。ｋ＝１６であり、ｎ＝２４であると仮定し、２４個のパケットの転送の後に、位置１、２、４、５、６、７、８、９、１１、１２、１３、１４、１５、１６、１８、２０、および２２のパケットが受信されたと仮定する。位置３および１０の組織的パケットが失われているが、他の１４個は正しく受信されている。したがって、q_1＝３およびq_2＝１０である。カウンタｌに０をセットし、２＋０＝２個の消去されない冗長パケットが得られる。これらは、位置１８および２０のパケットとすることができ、したがって、p_1＝１８およびp_2＝２０である。生成行列の行１８および２０、列３および１０の行列すなわち、行２および４、列３および１０のＣの部分行列をセット・アップする。この行列のランク（階数）が２ではなく、その結果、ステップ３（ｃ）が不成功であると仮定する。カウンタｌを１に増分し、ステップ３に戻って、処理を継続する。今回は、たとえばp_1＝１８、p_2＝２０、およびp_3＝２２をセットし、生成行列のこれらの行と列３および１０を含むＣの部分行列を形成することができる。この行列が、最大ランクである場合に、ステップ３（ｃ）が成功し、消去されていない組織的位置１８および２２に対応する行１および３が、可逆部分行列を形成する。ステップ３（ｄ）で、Ｂと呼ばれるこの行列の逆行列を計算し、

この行列と位置１８および２２の冗長パケットからなるデータにインターリーブ変換処理（ステップ３（ｅ））を適用することによって、位置３および１０の消去されたパケットの値が作られる。

【００８２】

いくつかの実施形態で、ｌの値を、ステップ２で０より大きいある数になるように選択することができる。たとえば、これは、ｌの小さい値についてステップ３（ｃ）が成功しない見込みがある場合に行うことができる。この開示を読んだ後に当業者が推測できる、多数の他の変形が可能である。

【００８３】

＜７．代数幾何符号（ａｌｇｅｂｒａｉｃ－ｇｅｏｍｅｔｒｉｃ　ｃｏｄｅ）＞

インターリーブ変換と組み合わされたサブシンボル復号の処理が特に良い結果を作る符号の１つの種類が、代数幾何符号または「ＡＧ符号」の種類である。ＡＧ符号は、同一の体の上のリードソロモン符号よりはるかに長い符号の構成を可能にする、リードソロモン符号の拡張である。これらの符号は、有限体上の曲線の点および指定された極点を有するその曲線上の関数を使用することによって構成される。これらの符号の構成は、有限の代数および符号化の理論の当業者に周知の概念であり、したがって、本明細書ではさらに詳細には説明しない。これらの符号に関する多数の文献の１つは、書籍、Ｇｏｐｐａ、Ｖ．Ｄ．、「Geometry and Codes」（Kluwer Academic Publishers 1988年）である。

【００８４】

ＡＧ符号は、リードソロモン符号の特性の多くを共有する。これらは、しばしば、明示的な生成行列およびパリティ検査行列によって記述され、その最小距離は、次元ｋおよびブロック長ｋを与えられれば、ｎ－ｋ＋１－ｇより小さくすることができず、ここで、ｇは、基礎になる曲線のパラメータである、非負の整数である。このパラメータを、曲線のジーナス（genus）と呼ぶ。ジーナス０の曲線は、本質的にリードソロモン符号を作り、より高いジーナスの曲線は、より小さい最小距離を犠牲にしても、ブロック長に関して実質的に改善された符号を作ることができる。

【００８５】

たとえば、基礎になる体がＧＦ（１６）である場合に、最長の可能なリードソロモン符号は、ブロック長１７を有する。対照的に、ブロック長２４を有するジーナス１のＡＧ符号を示すことが可能である。この符号の最小距離は、その最適値より１つ小さい。そのような符号を、次の例示的な事例で使用することができる。この事例は、例示のみのために示され、応用例の範囲を制限することを意図されたものではない。

【００８６】

１６ＫＢ長の１つのデータが、パケットが１ＫＢのペイロードを有するネットワークを介して転送されると仮定する。さらに、このデータが、３３％の消失に対して保護されなければならないと仮定する。その場合に、ジーナス１、ブロック長２４、および次元１６のＡＧ符号の生成行列を使用し、上で説明したインターリーブ変換を使用することによって、２４ＫＢの符号化されたコンテンツを作ることが可能である。この変換には、各サブシンボルがパケット・ペイロード（すなわち、シンボル）の１／４なので、サイズ２５６バイトのサブシンボルのＸＯＲをとることが含まれる。この結果の符号は、確率１で、受信された１７個のパケットの任意のセットから元の１６個のパケットを復号する能力を有し、約９６％の確率で１６個の受信されたパケットのセットから最初の１６個のパケットを復号する能力を有する（すなわち、１６個のパケットの可能な組合せのうちの９６％が、元の１６個のパケットがその組合せから復号可能である組合せである）。

【００８７】

ＡＧ符号は、組織的符号にすることができる。たとえば、上の場合に、符号化されたデータの最初の１６ＫＢを、元の１６ＫＢと等しいものにすることができ、追加の８ＫＢが、冗長データを表す。この冗長データを作るために、上で説明したインターリーブ変換が、８行１６列の行列に適用される。その場合に、冗長データを作るためのサブシンボルのＸＯＲの回数は、平均して８＊１６＊４／２すなわち２５６である。この動作の後に、作

られた符号化されたデータに、９６個のサブシンボルが含まれ、その結果、生成されるサブシンボルあたりのサブシンボルのＸＯＲの回数は、２５６／９６すなわち、３よりわずかに小さい。

【００８８】

上でリードソロモン符号が使用された場合には、リードソロモン符号が定義される、２のべき乗である最小の拡大体は、ＧＦ（２５６）である必要がある。その場合に、サブシンボルは、前の事例の半分の大きさになり、冗長データを作るのに、平均して８＊１６＊８／２＝５１２回のサブシンボルのＸＯＲが必要であり、これは、前の事例の半分の符号化速度になる。

【００８９】

下では、１ＫＢのパケット・ペイロード・サイズを有するパケットベース・ネットワークで６４ＫＢまでのサイズのコンテンツの転送に使用できる、いくつかのＡＧ符号を説明する。これらの例は、例示のみのために働き、本発明の範囲を制限するものと解釈されてはならない。これらの例は、元のコンテンツの長さではなく、符号化されたコンテンツの長さによってパラメータ化されている。後者への変換は、所望の保護消失率を介して行うことができる。たとえば、符号化されたコンテンツの長さが２４ＫＢであり、２５％消失の見込みがある場合に、元のコンテンツの長さを、１８ＫＢと等しくなるようにセットすることができる。

【００９０】

８ＫＢまでの符号化されたコンテンツサイズについて、９つの有理点という最大の可能な個数を有するＧＦ（４）上の楕円曲線からのＡＧ符号を使用することが可能である。そのような曲線の例が、エルミート曲線である。この曲線に対応する符号は、多くともさらに１つのパケットの追加によりコンテンツを回復することが可能であることを保証する。この作業のためのリードソロモン符号は、体ＧＦ（１６）で動作する必要があり、この例で構成されるＡＧ符号の約半分の符号化速度および復号速度を有する。

【００９１】

２４ＫＢまでの符号化されたコンテンツサイズについて、２５個の有理点という最大の可能な個数を有するＧＦ（１６）上の楕円曲線からのＡＧ符号を使用することが可能である。そのような曲線は、当業者に周知であり、対応する符号は、簡単に構成することができる。これらの符号について、上で説明した復号処理のインデックスｌが、絶対に１を超えないことが保証される。言い換えると、受信されたパケットの個数が、元のパケットの個数より１つ多い場合に、復号処理は成功する。しかし、受信されたパケットの個数が、元のパケットの個数と同一である場合には、復号器に関連する失敗の、ある確率がある。この確率は、この事例で数学的に計算することができ、おおむね１／２５すなわち４％と等しい。

【００９２】

３２ＫＢまでの符号化されたコンテンツサイズについて、３３個の点を有するＧＦ（１６）上のジーナス２の最大超楕円曲線（ｍａｘｉｍａｌ　ｈｙｐｅｒｅｌｌｉｐｔｉｃ　ｃｕｒｖｅ）からのＡＧ符号を使用することができる。この曲線も、当業者に周知であり、関連する符号の構成も当業者に周知である。この場合に、上の処理のインデックスｌは、絶対に２を超えない。３７ＫＢまでの符号化されたコンテンツサイズについて、３８個の点を有するＧＦ（１６）上のジーナス３の最大曲線（ｍａｘｉｍａｌ　ｃｕｒｖｅ）からのＡＧ符号を使用することができる。この場合に、上の処理のインデックスｌは、絶対に３を超えない。４４ＫＢまでの符号化されたコンテンツサイズについて、３８個の点を有するＧＦ（１６）上のジーナス４の最大曲線からのＡＧ符号を使用することができる。この場合に、上の処理のインデックスｌは、絶対に４を超えない。６４ＫＢまでの符号化されたコンテンツサイズについて、６５個の有理点を有するＧＦ（１６）上のエルミート曲線を使用することができる。この場合に、インデックスｌは、絶対に６を超えない。

【００９３】

上の事例のそれぞれで、ｋ個のパケットを含む元のコンテンツの回復は、任意のｋ個の

受信されたパケットから良い確率で可能であり、この確率は、受信されたパケットの個数がｋを超える際にすばやく増加し、追加が使用される曲線のジーナスと等しい場合に、１の確率に達する。

【００９４】

＜８．ランダム符号＞

インターリーブ変換と組み合わされたサブシンボル・ベース・符号化の処理は、決してブロック符号に固有ではない。一般に、本発明の教示は、有限体ＧＦ（２^{m}）上のすべての符号、またはより一般的に、すべての有限体ＧＦ（ｑ）上のすべての符号から利益を得ることができる。たとえば、この処理を、有益な効果のためにランダム符号と組み合わせることができる。

【００９５】

ランダム符号は、ブロック符号として、または連鎖反応符号に似た形で使用することができ、これに関して、生成できる出力符号の個数は、前もって固定されず、入力符号の個数より大きい大きさ程度とすることができる。具体的には、可能な出力符号の個数は、送信機（または、おそらくは調整されない、送信機のセット）が連続する転送に関して期待される時間の間に出力符号を繰り返さない、任意の期待される消失パターンより大きいものとすることができる。物理的処理が、真に無限で繰り返し可能であることはできないが、ルビーＩまたは他所に記載のように、所与の入力シーケンスの出力シンボルの個数が実質的に無限になる、連鎖反応符号を簡単に使用することができる。出力シンボルの関連しないシーケンスが、オーバーラップする可能性が非常に低い（所与の入力シーケンスに関する出力シンボルの大きい空間に起因して）ので、これらの符号を、時々、「情報加法的符号」と称する。

【００９６】

ランダム・ブロック符号について、生成行列は、ＧＦ（ｑ）の要素をランダムにまたは擬似ランダムに選択することによって得られる。本明細書で使用される「ランダム」に、「擬似ランダム」も含まれる場合があるが、開示の可読性を改善するために、これが、他所では明示的に述べられないことを理解されたい。体のサイズｑは、行列のランク特性に依存する。一般に、ｑが大きいほど、所与の次元の行列が最大ランクを有する可能性が高くなる。上で説明した復号処理のステップ３（ｃ）の成功は、行列のランク特性によって決定される。

【００９７】

ＧＦ（ｑ）上のランダム連鎖反応符号について、各出力シンボルは、ルビーＩおよびショクロラヒＩに記載のように、キーを使用して生成される。キーごとに、入力シンボルのサブセットが、選択された入力シンボル要素ごとの体ＧＦ（ｑ）内のランダムなまたは擬似ランダムな要素と一緒に選択される。概念上、ＧＦ（ｑ）内の選択された値および選択されなかった入力シンボルに対応する位置の０を含むベクトルが形成され、インターリーブ変換と組み合わされたサブシンボル・ベース・符号化の処理が、この行列に適用される。この行列を作成する中間ステップは、純粋に概念的であり、応用例では完全に省略することができる。

【００９８】

一例として、ＧＦ（１６）上の符号の事例をもう一度検討されたい。ｋ行ｋ列のランダム行列は、約９３％と等しい確率で、ＧＦ（１６）上で可逆である（本明細書で何かが「ランダム」と記述される場合に、必ず、そうでないと示されない限り、「擬似ランダム」もあてはまることを理解されたい）。これは、上で説明した復号処理を適用する際に、事例のうちの９３％で、カウンタｌが０のままになり、したがって、元のデータのサイズを超えるデータの受信が不要であることを意味する。ｋ行ｋ＋１列のランダム行列は、約９９．５％と等しい確率でランクｋを有する。これは、カウンタｌが、事例のうちの０．５％に限って２に達することを意味する。同様に、ｌが絶対に２を超えない確率は、約９９．９７％であり、ｌが絶対に３を超えない確率は、約９９．９９８％であり、ｌが絶対に４を超えない確率は、約９９．９９９９８％であり、以下同様であり、ｌが６を超える確

率は、約４×１０$^{-9}$である。表１に、受信された追加パケットの個数のさまざまな他の値に関する、復号器の誤り確率を示す。表１のデータは、非常に良いオーバーヘッドを有し、より小さい体の上で作られるのでリードソロモン符号より効率的な符号化アルゴリズムおよび復号アルゴリズムを有するランダム符号を構成することが可能であることを示す。

【００９９】

【表１】

TABLE 1.

追加受信パケット	復号誤り確率
0	6.6×10^{-2}
1	4.2×10^{-3}
2	2.7×10^{-4}
3	1.7×10^{-5}
4	1.2×10^{-6}
5	6.4×10^{-8}
6	4.0×10^{-9}
7	2.5×10^{-10}
8	1.6×10^{-11}
9	9.8×10^{-13}
10	6.1×10^{-14}

【０１００】

類似する結果が、体ＧＦ（４）で発生する。表２に、ＧＦ（４）に関する、受信された追加パケットの個数の関数としての復号器の誤り確率を示す。ＧＦ（４）が、ＧＦ（１６）上の符号と比較して２倍だけ符号化速度および復号速度を高めることに留意されたい。

【０１０１】

【表２】

TABLE 2.

追加受信パケット	復号誤り確率
0	3.1×10^{-1}
1	8.2×10^{-2}
2	2.1×10^{-2}
3	5.2×10^{-3}
4	1.3×10^{-3}
5	3.3×10^{-4}
6	8.2×10^{-5}
7	2.1×10^{-5}
8	5.1×10^{-6}
9	1.3×10^{-6}
10	3.2×10^{-7}

【０１０２】

上の数字は、ＧＦ（４）またはＧＦ（１６）上のランダム符号に基づく情報加法的符号が、非常に小さいオーバーヘッドであっても優秀に実行することを暗示する。

【０１０３】

複数の例を、インターリーブ変換と共に使用される非リードソロモン・符号化について示したが、他の非リードソロモン・ベース行列も動作することができる。

【０１０４】

＜変形形態＞

いくつかの変形形態で、符号器は、出力シンボルをよりすばやく生成するために並列に

動作する。サブシンボル動作のある利益を得るために、並列符号器モジュールは、完全に独立に動作するのではなく、相互依存でなければならない。たとえば、並列符号器は、サブシンボルが複数の入力シンボルにまたがって、入力シンボル内の異なるサブシンボル位置から混合されるように、値関数の適用に関して複数の入力シンボルにまたがってサブシンボル・セットの選択を調整する。

【０１０５】

＜結論＞

リードソロモン・符号より少ない数の算術演算を用いて動作できるサブシンボル・ベース・符号化について述べた。本発明人は、最適性条件が必要である場合に、そのような符号が存在しないが、その要件を緩和することによって、興味ある符号が可能になることに気付いた。そのような符号について、元のコンテンツを復号できるようになるために受信する必要がある出力シンボルの個数に関して最適性条件を失わなければならないことを示すことができるが、ある種類の符号は、ほとんどの場合にリードソロモン符号に似て実行することを示す穏当な統計的特性を示し、ごく少数の場合に、元のコンテンツを回復するために余分のシンボルを必要とする。

【０１０６】

絶対的最適性が、必ずしも要件ではなく、いずれにせよ必ずしもデータの完全な回復にはつながらないことの観察から、十分に良い最適に近い転送効率を、しばしば、かなり少ない計算労力で有することができる。たとえば、より小さいアルファベットを用いる符号を使用することによって、計算労力が大幅に減ると同時に、絶対最適性からのごくわずかな緩和だけが引き起こされる。

【０１０７】

いくつかの好ましい実施形態で、上で説明した通信処理を実行する命令のセット（またはソフトウェア）が、おそらく損失のある通信媒体を介して通信する複数の多目的計算機械に提供される。機械の台数は、１つの送信機および１つの受信機から、任意の台数の送信し、かつ／または受信する機械までの範囲とすることができる。機械を接続する通信媒体は、有線、光、無線、または類似物とすることができる。上で説明した通信システムは、多数の用途を有し、これらの用途は、この説明から明白である。

【０１０８】

上の説明は、例示的であって制限的ではない。本発明の多数の変形形態が、この開示を再検討した時に当業者に明白になるであろう。したがって、本発明の範囲は、上の説明を参照することによって決定されてはならず、その代わりに、添付請求項およびその同等物の全範囲を参照することによって決定されなければならない。

【図面の簡単な説明】

【０１０９】

【図１】本発明の一実施形態による通信システムを示すブロック図である。

【図２】図１の符号器の一部を詳細に示すブロック図である。

【図３Ａ】ＧＦ（４）上の基礎行列を示す。

【図３Ｂ】ＧＦ（２）上の２進生成行列を示す。

【図４】図１の出力シンボル生成器を示す図である。

【図５】図１のサブシンボル生成器を示す図である。

【図１】

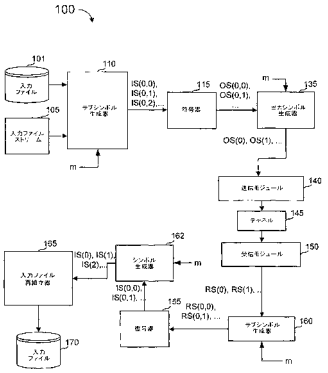

Figure 1

【図２】

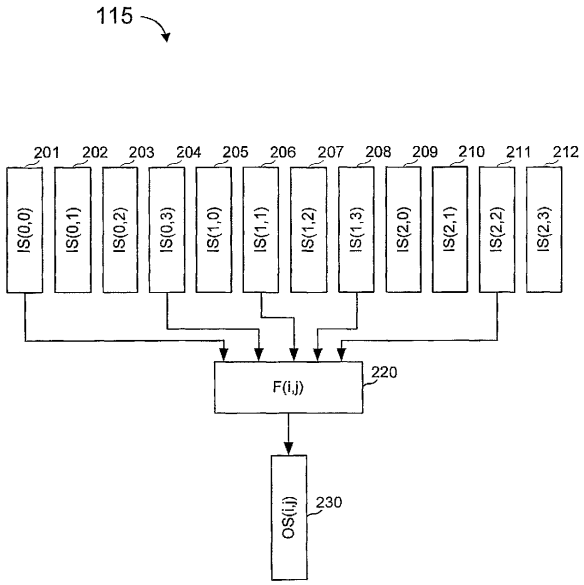

Figure 2

【図３Ａ】

0	1	α	1	α^2
1	0	1	α	α

Figure 3A

【図３Ｂ】

0	0	1	0	0	1	1	0	1	1
0	0	0	1	1	1	0	1	1	0
1	0	0	0	1	0	0	1	0	1
0	1	0	0	0	1	1	1	1	1

Figure 3B

【図４】

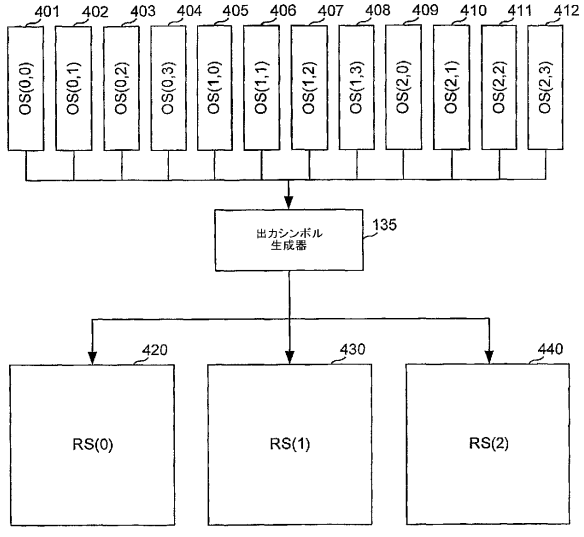

Figure 4

【図５】

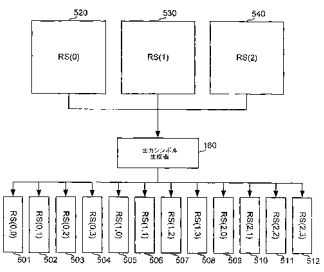

Figure 5

フロントページの続き

(74)代理人 100075672
弁理士 峰 隆司
(74)代理人 100095441
弁理士 白根 俊郎
(74)代理人 100084618
弁理士 村松 貞男
(74)代理人 100103034
弁理士 野河 信久
(74)代理人 100119976
弁理士 幸長 保次郎
(74)代理人 100153051
弁理士 河野 直樹
(74)代理人 100140176
弁理士 砂川 克
(74)代理人 100100952
弁理士 風間 鉄也
(74)代理人 100101812
弁理士 勝村 紘
(74)代理人 100070437
弁理士 河井 将次
(74)代理人 100124394
弁理士 佐藤 立志
(74)代理人 100112807
弁理士 岡田 貴志
(74)代理人 100111073
弁理士 堀内 美保子
(74)代理人 100134290
弁理士 竹内 将訓
(74)代理人 100127144
弁理士 市原 卓三
(74)代理人 100141933
弁理士 山下 元
(72)発明者 ショクロラヒ, エム. , アミン
アメリカ合衆国 カリフォルニア州 ９５１２３, サン ホセ, チャンドラー コート ５７８０

審査官 渡辺 未央子

(56)参考文献 特開２００２－２０４２１９（ＪＰ，Ａ）
特開２００４－０４８７０４（ＪＰ，Ａ）

(58)調査した分野(Int.Cl., ＤＢ名)
H03M 13/15