

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4506185号
(P4506185)

(45) 発行日 平成22年7月21日(2010.7.21)

(24) 登録日 平成22年5月14日(2010.5.14)

(51) Int. Cl.		F I			
HO4L	1/00	(2006.01)	HO4L	1/00	B
GO6F	11/10	(2006.01)	GO6F	11/10	33OM
HO3M	13/19	(2006.01)	HO3M	13/19	

請求項の数 7 (全 50 頁)

(21) 出願番号	特願2004-30402 (P2004-30402)	(73) 特許権者	000002185
(22) 出願日	平成16年2月6日(2004.2.6)		ソニー株式会社
(65) 公開番号	特開2005-223684 (P2005-223684A)		東京都港区港南1丁目7番1号
(43) 公開日	平成17年8月18日(2005.8.18)	(74) 代理人	100082131
審査請求日	平成19年1月26日(2007.1.26)		弁理士 稲本 義雄
		(72) 発明者	小林 稔治
			東京都品川区北品川6丁目7番35号 ソニー株式会社内
		審査官	谷岡 佳彦

最終頁に続く

(54) 【発明の名称】 受信装置および方法、並びにプログラム

(57) 【特許請求の範囲】

【請求項1】

送信装置が送信する送信用データのブロックデータが分割されたパケットデータを受信する受信装置であって、

前記パケットデータである第1のデータ、並びに、複数の前記第1のデータの排他的論理和演算結果である第2のデータの両方を受信する受信手段と、

前記受信手段により受信された前記第2のデータに付加されているパケット番号順に前記第2のデータを整列させ整列手段と、

所定の係数行列から、前記受信手段により受信された前記第1のデータに付加されているパケット番号と、前記受信手段により受信された前記第2のデータに付加されているパケット番号とに対応する第1の部分行列を取得する第1の取得手段と、

前記第1の取得手段により取得された前記第1の部分行列を用いて、計算用の前記第2のデータを算出するための第1の行列式を作成する第1の行列式作成手段と、

前記第1の行列式作成手段により作成された前記第1の行列式を演算し、前記計算用の第2のデータを算出する算出手段と、

前記係数行列から、前記受信手段により受信できなかった前記第1のデータのパケット番号と、前記受信手段により受信された前記第2のデータのパケット番号とに対応する第2の部分行列を取得する第2の取得手段と、

前記第2の取得手段により取得された前記第2の部分行列を用いて、前記計算用の第2のデータを算出するための第2の行列式を作成する第2の行列式作成手段と、

10

20

前記第 2 の行列式作成手段により作成された前記第 2 の行列式を解くことにより、前記受信手段により受信できなかった前記第 1 のデータを復元する復元手段と
を備える受信装置。

【請求項 2】

所定の種データを用いて前記係数行列を生成する生成手段をさらに備える
請求項 1 に記載の受信装置。

【請求項 3】

前記生成手段は、前記送信装置と同一の前記所定の種データを用いて、前記送信装置が
前記第 2 のデータを生成するために生成した係数行列と同一の係数行列を生成する
請求項 2 に記載の受信装置。

10

【請求項 4】

前記生成手段は、前記送信装置から送信される前記所定の種データを用いて前記係数行
列を生成する
請求項 2 に記載の受信装置。

【請求項 5】

前記生成手段は、前記受信手段により受信された前記第 2 のデータの packets 番号を前
記種データとして前記係数行列を生成する
請求項 2 に記載の受信装置。

【請求項 6】

送信装置が送信する送信用データのブロックデータが分割された packets データを受信
する受信装置の受信方法であって、

20

前記受信装置の受信手段が、前記 packets データである第 1 のデータ、並びに、複数の
前記第 1 のデータの排他的論理和演算結果である第 2 のデータの両方を受信し、

前記受信装置の整列手段が、受信された前記第 2 のデータに付加されている packets 番
号順に前記第 2 のデータを整列させ、

前記受信装置の第 1 の取得手段が、所定の係数行列から、受信された前記第 1 のデータ
に付加されている packets 番号と、受信された前記第 2 のデータに付加されているパケッ
ト番号とに対応する第 1 の部分行列を取得し、

前記受信装置の第 1 の行列式作成手段が、取得された前記第 1 の部分行列を用いて、計
算用の前記第 2 のデータを算出するための第 1 の行列式を作成し、

30

前記受信装置の算出手段が、作成された前記第 1 の行列式を演算し、前記計算用の第 2
のデータを算出し、

前記受信装置の第 2 の取得手段が、前記係数行列から、受信できなかった前記第 1 のデ
ータの packets 番号と、受信された前記第 2 のデータの packets 番号とに対応する第 2 の
部分行列を取得し、

前記受信装置の第 2 の行列式作成手段が、取得された前記第 2 の部分行列を用いて、前
記計算用の第 2 のデータを算出するための第 2 の行列式を作成し、

前記受信装置の復元手段が、作成された前記第 2 の行列式を解くことにより、受信でき
なかった前記第 1 のデータを復元する

受信方法。

40

【請求項 7】

送信装置が送信する送信用データのブロックデータが分割された packets データを受信
するコンピュータを、

前記 packets データである第 1 のデータ、並びに、複数の前記第 1 のデータの排他的論
理和演算結果である第 2 のデータの両方を受信する受信手段と、

前記受信手段により受信された前記第 2 のデータに付加されている packets 番号順に前
記第 2 のデータを整列させ整列手段と、

所定の係数行列から、前記受信手段により受信された前記第 1 のデータに付加されてい
る packets 番号と、前記受信手段により受信された前記第 2 のデータに付加されているパ
ckets 番号とに対応する第 1 の部分行列を取得する第 1 の取得手段と、

50

前記第 1 の取得手段により取得された前記第 1 の部分行列を用いて、計算用の前記第 2 のデータを算出するための第 1 の行列式を作成する第 1 の行列式作成手段と、

前記第 1 の行列式作成手段により作成された前記第 1 の行列式を演算し、前記計算用の第 2 のデータを算出する算出手段と、

前記係数行列から、前記受信手段により受信できなかった前記第 1 のデータの packets 番号と、前記受信手段により受信された前記第 2 のデータの packets 番号とに対応する第 2 の部分行列を取得する第 2 の取得手段と、

前記第 2 の取得手段により取得された前記第 2 の部分行列を用いて、前記計算用の第 2 のデータを算出するための第 2 の行列式を作成する第 2 の行列式作成手段と、

前記第 2 の行列式作成手段により作成された前記第 2 の行列式を解くことにより、前記受信手段により受信できなかった前記第 1 のデータを復元する復元手段

10

として機能させるプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、受信装置および方法、並びにプログラムに関し、特に、パケットロスの復元処理の負荷を軽減させ、より正確な通信を、より容易に行うことができるようにした受信装置および方法、並びにプログラムに関する。

【背景技術】

【0002】

20

従来、データ通信における誤り訂正方法として、様々な方法が提案されている。例えば、広い帯域に分散するレベルの大きい雑音が重畳するような有線、無線、光等の伝送媒体や記録媒体を介した複数チャネルのデータに対して、複数チャネルの中の少なくとも 1 チャネルをエラー訂正チャネルとし、他のチャネルの排他的論理和又はモジュロ加算の結果をエラー訂正チャネルのデータとして、受信判定した複数チャネルのデータ又は再生判定した複数チャネルのデータについて、エラー訂正チャネルの判定データと、そのエラー訂正チャネルと組としたチャネルの判定データとの排他的論理和又はモジュロ加算の結果が“0”等の予め定めた特定の値でない場合にエラー発生と判定し、且つチャネル毎の品質を検出して品質劣化のチャネルのデータのエラー発生確率が高いので、このチャネルのデータを除く他のチャネルのデータの排他的論理和又はモジュロ加算の結果を、品質劣化の

30

【0003】

データ通信としては、近年、ブロードバンド通信網の発達などにより、インターネットに代表されるネットワークを介して音声や映像等のデータのストリーミング配信やリアルタイムな放送等が数多く行われるようになってきた。

【0004】

このようなネットワークを介したデータ配信においては、ネットワークにおいてパケットロスが発生する場合があります。上述したような受信したデータに対するエラー訂正だけでは対処できない場合があった。

40

【0005】

そこで、ロスしたパケットに対するエラー訂正方法として、例えば、受信側が、ロスしたパケットの再送を送信側に要求する再送制御を行う方法やリードソロモン符号を用いた誤り検出訂正方法 (FEC: Forward Error Correction) が提案されている (例えば、特許文献 2 参照)。

【0006】

リードソロモン符号によるパケットロスに対するエラー訂正処理において、送信装置は、所定の大きさのブロック単位ごとに、送信データの誤り訂正コードを生成し、その誤り訂正コードより冗長パケットを生成し、その生成した冗長パケットをパケット化した元のデータとともに配信する。

50

【 0 0 0 7 】

受信装置は、このように配信された元のデータの packets および冗長 packets を受信する。そして、元のデータの packets をロスした場合は、受信した冗長 packets を用いて、そのロスした packets を復元する。

【 0 0 0 8 】

【特許文献 1】特許 3 4 2 7 3 8 2 号

【 0 0 0 9 】

【特許文献 2】特開平 1 1 - 1 3 6 2 2 0 号公報

【発明の開示】

【発明が解決しようとする課題】

10

【 0 0 1 0 】

しかしながら、上述した再送制御を行う方法においては、packet ロスが発生する度に、受信側から送信側に対して再送要求が供給されるため、高い確率でエラー訂正を行うことができるが、例えば、マルチキャストのように、複数の装置に対してデータを送信する場合、受信側の各装置より供給される再送要求が、送信側の装置に集中し、送信側の装置の負荷が増大する恐れがある。また、このような再送制御は、通信を伴うため、エラー訂正処理に所定の時間が必要になり、例えば、ストリーミング配信のように、受信側の再生処理にリアルタイム性が求められる場合、処理が間に合わない恐れがあった。

【 0 0 1 1 】

また、上述したようなリードソロモン符号を用いた誤り検出訂正方法の場合、冗長データの作成処理やロス packets の復元処理において、ガロア体上の演算が必要であり、送受信処理の負荷が増大してしまうという課題があった。

20

【 0 0 1 2 】

従って、例えば、コンテンツの配信サーバが、マルチキャストやブロードキャスト等のように、多数の端末装置に対してコンテンツデータをストリーミング配信する際に、上述したようなリードソロモン符号を用いた誤り検出訂正方法を適用する場合、ストリーミング配信のために配信処理にリアルタイム性が求められるので、配信サーバには、負荷の高い冗長 packet 生成処理を高速で実行するような高性能なハードウェアスペックが要求されるので、製造コストが増大してしまう。同様に、復元処理を実行する端末装置も、高性能なハードウェアスペックが要求され、製造コストが増大してしまう。

30

【 0 0 1 3 】

そこで、製造コストが下がるように、ガロア体上の演算処理を減らすために、生成する冗長 packet を削減する方法が考えられるが、冗長 packet を削減すると、端末装置において、packet の復元性能が低下してしまい、エラー訂正ができない場合は packet の再送を要求する必要が生じる場合がある。

【 0 0 1 4 】

しかしながら、マルチキャストやブロードキャストの場合、配信サーバが配信を行いながら各端末装置からの再送要求を全て処理すると、負荷が増大し、配信処理に影響を与えてしまう恐れがあった。さらに、ストリーミング配信の場合、その再送要求にもリアルタイム性を求められるため、再送要求に正常に回答するのは事実上困難であり、ストリーミング配信を正常に行うことができなくなってしまう恐れがあった。

40

【 0 0 1 5 】

本発明は、このような状況に鑑みてなされたものであり、復元処理の負荷を軽減させることにより、高速であり、かつ、復元性能の高い復元処理を、製造コストを増大せず実現し、より容易に、より正確な通信を行うことができるようにするものである。

【課題を解決するための手段】

【 0 0 3 1 】

本発明の受信装置は、packet データである第 1 のデータ、並びに、複数の第 1 のデータの排他的論理和演算結果である第 2 のデータの両方を受信する受信手段と、前記受信手段により受信された前記第 2 のデータに付加されている packet 番号順に前記第 2 のデー

50

データを整列させ整列手段と、所定の係数行列から、前記受信手段により受信された前記第1のデータに付加されているパケット番号と、前記受信手段により受信された前記第2のデータに付加されているパケット番号とに対応する第1の部分行列を取得する第1の取得手段と、前記第1の取得手段により取得された前記第1の部分行列を用いて、計算用の前記第2のデータを算出するための第1の行列式を作成する第1の行列式作成手段と、前記第1の行列式作成手段により作成された前記第1の行列式を演算し、前記計算用の第2のデータを算出する算出手段と、前記係数行列から、前記受信手段により受信できなかった前記第1のデータのパケット番号と、前記受信手段により受信された前記第2のデータのパケット番号とに対応する第2の部分行列を取得する第2の取得手段と、前記第2の取得手段により取得された前記第2の部分行列を用いて、前記計算用の第2のデータを算出するための第2の行列式を作成する第2の行列式作成手段と、前記第2の行列式作成手段により作成された前記第2の行列式を解くことにより、前記受信手段により受信できなかった前記第1のデータを復元する復元手段とを備えることを特徴とする。

10

【0032】

所定の種データを用いて前記係数行列を生成する生成手段をさらに備えることができる。

【0033】

前記生成手段は、前記送信装置と同一の前記所定の種データを用いて、前記送信装置が前記第2のデータを生成するために生成した係数行列と同一の係数行列を生成することができる。

20

【0034】

前記生成手段は、前記送信装置から送信される前記所定の種データを用いて前記係数行列を生成することができる。

【0035】

前記生成手段は、前記受信手段により受信された前記第2のデータのパケット番号を前記種データとして前記係数行列を生成することができる。

【0036】

本発明の受信方法は、送信装置が送信する送信用データのブロックデータが分割されたパケットデータを受信する受信装置の受信方法であって、前記受信装置の受信手段が、前記パケットデータである第1のデータ、並びに、複数の前記第1のデータの排他的論理和演算結果である第2のデータの両方を受信し、前記受信装置の整列手段が、受信された前記第2のデータに付加されているパケット番号順に前記第2のデータを整列させ、前記受信装置の第1の取得手段が、所定の係数行列から、受信された前記第1のデータに付加されているパケット番号と、受信された前記第2のデータに付加されているパケット番号とに対応する第1の部分行列を取得し、前記受信装置の第1の行列式作成手段が、取得された前記第1の部分行列を用いて、計算用の前記第2のデータを算出するための第1の行列式を作成し、前記受信装置の算出手段が、作成された前記第1の行列式を演算し、前記計算用の第2のデータを算出し、前記受信装置の第2の取得手段が、前記係数行列から、受信できなかった前記第1のデータのパケット番号と、受信された前記第2のデータのパケット番号とに対応する第2の部分行列を取得し、前記受信装置の第2の行列式作成手段が、取得された前記第2の部分行列を用いて、前記計算用の第2のデータを算出するための第2の行列式を作成し、前記受信装置の復元手段が、作成された前記第2の行列式を解くことにより、受信できなかった前記第1のデータを復元する。

30

40

【0037】

本発明のプログラムは、送信装置が送信する送信用データのブロックデータが分割されたパケットデータを受信するコンピュータを、前記パケットデータである第1のデータ、並びに、複数の前記第1のデータの排他的論理和演算結果である第2のデータの両方を受信する受信手段と、前記受信手段により受信された前記第2のデータに付加されているパケット番号順に前記第2のデータを整列させ整列手段と、所定の係数行列から、前記受信手段により受信された前記第1のデータに付加されているパケット番号と、前記受信手段

50

により受信された前記第2のデータに付加されているパケット番号とに対応する第1の部分行列を取得する第1の取得手段と、前記第1の取得手段により取得された前記第1の部分行列を用いて、計算用の前記第2のデータを算出するための第1の行列式を作成する第1の行列式作成手段と、前記第1の行列式作成手段により作成された前記第1の行列式を演算し、前記計算用の第2のデータを算出する算出手段と、前記係数行列から、前記受信手段により受信できなかった前記第1のデータのパケット番号と、前記受信手段により受信された前記第2のデータのパケット番号とに対応する第2の部分行列を取得する第2の取得手段と、前記第2の取得手段により取得された前記第2の部分行列を用いて、前記計算用の第2のデータを算出するための第2の行列式を作成する第2の行列式作成手段と、前記第2の行列式作成手段により作成された前記第2の行列式を解くことにより、前記受信手段により受信できなかった前記第1のデータを復元する復元手段として機能させる。

10

【0043】

本発明の受信装置および方法、並びにプログラムにおいては、パケットデータである第1のデータ、並びに、複数の第1のデータの排他的論理和演算結果である第2のデータの両方が受信され、受信された第2のデータに付加されているパケット番号順に第2のデータが整列させられ、所定の係数行列から、受信された第1のデータに付加されているパケット番号と、受信された第2のデータに付加されているパケット番号とに対応する第1の部分行列が取得され、取得された第1の部分行列を用いて、計算用の第2のデータを算出するための第1の行列式が作成され、作成された第1の行列式が演算され、計算用の第2のデータが算出され、係数行列から、受信できなかった第1のデータのパケット番号と、

20

受信された第2のデータのパケット番号とに対応する第2の部分行列が取得され、取得された第2の部分行列を用いて、計算用の第2のデータを算出するための第2の行列式が作成され、作成された第2の行列式を解くことにより、受信できなかった第1のデータが復元される。

【発明の効果】

【0045】

本発明によれば、データを送受信することができる。特に、より高性能な復元処理を行うことができ、より正確な通信を、より容易に行うことができる。

【発明を実施するための最良の形態】

【0073】

以下に、本発明の実施の形態について図面を参照して説明する。

30

【0074】

図1は、本発明を適用した通信システムの構成例を表している。

【0075】

図1の通信システム1は、データを送信する送信装置11、LAN(Local Area Network)やインターネット等に代表されるネットワークであり、送信装置11が接続されるネットワーク12、および、ネットワーク12に接続され、送信装置11より送信されるデータを受信する受信装置13により構成される。

【0076】

送信装置11は、外部より供給されたストリームデータ等の、送信用のコンテンツデータ(送信用データ)を取得すると、その送信用データを、ネットワーク12を介して、データの送信先として設定されている受信装置13に送信する。その際、送信装置11は、送信用データを分割してパケット化し、UDP(User Datagram Protocol)等の所定のプロトコルを用いて各パケットを送信する。また、送信装置11は、後述するように、送信用データからデータ復元処理用の冗長データを生成し、その冗長データもパケット化し、元のデータ(送信用データ)とともに受信装置13に送信する。

40

【0077】

受信装置13は、送信装置11より供給されるパケットを受信すると、受信したパケットに基づいて、元の送信用データ(コンテンツデータ)を復元し、復元したコンテンツデータを外部に供給する。その際、受信装置13は、受信できなかった(ロスした)パケッ

50

ト（送信用データの packets）を、受信した冗長データを用いて復元する。

【0078】

以上のように、通信システム1において、送信装置11は、送信用データの他に、その送信用データより生成した、パケットロスに対する誤り訂正用の冗長データを複数生成し、それらを送信用データとともに受信装置13に送信する。受信装置13は、その冗長データ（受信した冗長データ）に基づいて、受信できなかった元の送信用データの復元処理を行う。このようにすることにより、受信装置13は、送信装置11に再送要求を行うことなく、ロスしたデータを復元することができる。

【0079】

図2は、送信装置11の詳細な構成例を示すブロック図である。なお、送信装置11により処理される送信用データの変化を、図3を用いて説明する。

10

【0080】

例えば、図2において、送信装置11のデータ分割部31は、送信装置11の外部より供給されたコンテンツデータ等の送信データを所定のデータサイズのブロックに分割し、それをブロック分割部32に供給する。すなわち、データ分割部31は、供給される送信データを内蔵するバッファ（図示せず）に蓄積していき、所定のデータサイズのデータが蓄積される度に、それをブロックとしてブロック分割部32に供給する。

【0081】

例えば、図3において、送信装置11に供給される送信データ51は、データ分割部31により、所定のデータサイズ単位の複数のブロック52に分割される。図3の場合、送信データ51は3個のブロック52に分割される。

20

【0082】

ブロック分割部32は、データ分割部31と同様にして、データ分割部31より供給されるブロック単位の送信データを、さらに所定のデータサイズに分割する。すなわち、ブロック分割部32は、ブロック単位の送信データを、後段において、パケット化する際のデータサイズ（パケットサイズ）に分割し、パケット用のデータ（パケットデータ）を生成する。パケットデータを生成したブロック分割部32は、それを、冗長データ生成部33およびアプリケーションデータ生成部34に供給する。

【0083】

例えば、図3において、ブロック52は、それぞれ、ブロック分割部32により、パケット化する際のデータサイズのパケットデータである元データ53に分割される。図3の場合、各ブロックはそれぞれ6個の元データ61に分割される。もちろん、この分割数は、パケットデータのデータサイズとブロックのデータサイズによって決定されるものであり、1つのブロックをいくつのパケットデータに分割するようにしてもよい。

30

【0084】

冗長データ生成部33は、元の送信用データを分割したパケットデータ（元データ）を1ブロック分取得すると、その元データに基づいて、受信側においてロスしたパケットを復元するために用いられる冗長データを生成する。具体的には、冗長データ生成部33は、1ブロック分の元データより、任意の元データを選択し、それらをEOR演算（Exclusive OR：排他的論理和）により合成し、冗長データとする。より具体的な生成方法については後述する。

40

【0085】

なお、この復元処理用の冗長データは、EOR演算により複数の元データを合成したものであり、そのまま元データとして利用することはできないが、実質、元データの内容を含んでいることになる。従って、元データと情報が重複しているので、この復元処理用のデータのことを「冗長データ」と称する。また、この冗長データは、後述するように、元データと同じデータサイズであり、この冗長データもパケット化されて送信される。

【0086】

すなわち、パケットデータ（パケットサイズのデータ）には、元データ（送信用データを分割して生成されたパケットデータ）と、冗長データ（元データに基づいて作成された

50

ロス packets 復元用の packets データ)とがあり、以下においては、元データと冗長データを区別する必要がない場合は、packets データと称し、区別する必要がある場合は、それぞれを元データ、冗長データと称する。

【0087】

冗長データ生成部 33 は、生成した冗長データをアプリケーションデータ生成部 34 に供給する。

【0088】

アプリケーションデータ生成部 34 は、ブロック分割部 32 より供給された元データ、および、冗長データ生成部 33 より供給された冗長データに、それぞれ、ヘッダを付加し、アプリケーションデータを生成する。すなわち、アプリケーションデータには、元データにヘッダが付加されたものと、冗長データにヘッダが付加されたものがある。ヘッダの内容については、後述する。

10

【0089】

アプリケーションデータ生成部 34 は、生成したアプリケーションデータを UDP packets 化部 35 に供給する。

【0090】

なお、ここまでの処理は、ISO (International Organization for Standardization) が制定した 7 階層のネットワークプロトコルの構造のモデルである OSI 階層モデル (Open Systems Interconnection layer model) における、アプリケーション層、プレゼンテーション層、およびセッション層の処理である。

20

【0091】

アプリケーションデータを供給された UDP packets 化部 35 は、そのアプリケーションデータを UDP プロトコルに基づいて packets 化し、その生成した UDP packets をバッファ 36 に供給する。

【0092】

図 4 に示されるように、UDP packets 61 は、64 ビットの UDP ヘッダ 62 およびアプリケーションデータ 63 により構成される。UDP ヘッダ 62 には、送信元のポート番号、送信先のポート番号、アプリケーションデータ 63 のデータ長、およびチェックサム等の情報が含まれる。アプリケーションデータのデータサイズは、任意に定められた所定の値であるが、通常 1 K バイト程度である。

30

【0093】

バッファ 36 は、RAM (Random Access Memory) 等の半導体メモリ等により構成され、UDP packets 化部 35 より供給される UDP packets を蓄積し、出力制御部 37 の制御に基づいて、その蓄積している UDP packets を出力する。

【0094】

出力制御部 37 は、バッファ 36 に UDP packets が 1 ブロック分蓄積されると、それらの UDP packets を任意の (ランダムな) 順番で取得し、それらを IP (Internet Protocol) 処理部 38 に供給する。すなわち、バッファ 36 および出力制御部 37 は、UDP packets 化部 35 において生成された UDP packets の出力順を変更 (シャッフル) し、元データを含む UDP packets と、冗長データを含む UDP packets を任意の順番で出力させる。

40

【0095】

一般に、packets の送信時にネットワーク等において、受信側が連続する複数の packets をバースト的にロスする (受信できない) 場合がある。また、後述するように、受信装置 13 において、ロスした元データを復元する際の復元能力は、受信した冗長データの個数に約比例する。従って、送信装置 11 が元データと冗長データをそれぞれ順番に出力した場合、そのようなバースト的 packets ロスにより、packets のロスが冗長データに集中してしまうと、復元能力が極端に低下する恐れがある。

【0096】

そこで、送信装置 11 のバッファ 36 および出力制御部 37 が、以上のように UDP packets の出力順をシャッフルして、元データと冗長データを混在させた順番で送信するよう

50

にすることにより、パケットの送信時にバースト的パケットロスが発生しても、そのロスが冗長データに集中することを抑制することができ、バースト的パケットロスに対する耐性を強くすることができる。すなわち、送信装置 11 は、より正確にデータを受信装置 13 に送信することができ、通信システム 1 は、より正確なデータの送受信を実現することができる。

【0097】

例えば、図 3 において、元データ 53 および冗長データ 54 は UDP パケット化された後、その出力順がシャッフルされて、送信パケット 55 として任意の順番で送信される。

【0098】

なお、これらの UDP パケット化部 35 乃至出力制御部 37 による処理は、OSI 階層モデルにおける、トランスポート層の処理である。

10

【0099】

IP 処理部 38 は、OSI 階層モデルにおける、ネットワーク層の処理を行い、IP プロトコルに基づいて、出力制御部 37 より供給される UDP パケットをそれぞれ IP パケット化し、イーサネット(R) (Ethernet(R)) 処理部 39 に供給する。

【0100】

イーサネット(R) 処理部 39 は、イーサネット(R) の規格に基づいて、データリンク層および物理層の処理を行い、IP 処理部 38 より供給された IP パケットをさらにパケット化し、受信装置 13 に送信するように、それを、ケーブル 40 を介してネットワーク 12 に出力する。

20

【0101】

イーサネット(R) 処理部 39 より出力されたパケットは、ネットワーク 12 を介して受信装置 13 に供給される。

【0102】

次に、図 2 の冗長データ生成部 33 の詳細について説明する。なお、以下においては、送信装置 11 は、1 ブロックの送信データより、N 個の元データ D (D_1 乃至 D_N) を生成し、その N 個の元データ D より M 個の冗長データ S (S_1 乃至 S_M) を生成する場合について説明する。図 5 は、冗長データ生成部 33 の詳細な構成例を示すブロック図である。

【0103】

図 5 において、図 2 のブロック分割部 32 より供給された元データは、元データ保持部 71 に供給され、保持される。元データ保持部 71 は、1 ブロック分の元データを保持すると、その元データを行列式生成部 72 に供給する。

30

【0104】

1 ブロック分の元データを取得すると、行列式生成部 72 は、係数行列供給部 73 に対して、冗長データを生成するための行列式の生成に用いられる係数行列を要求する(図中、点線矢印)。係数行列は、全成分が「1」または「0」の M 行 N 列の行列であり、EOR 演算を行う元データを選択するための行列である。後述するように、全冗長データは、係数行列と、各元データを成分とする行列との積算により求められる。従って、係数行列の各行は、互いに異なる冗長データに対応し、その各行の各成分は、それぞれ互いに異なる元データと積算される。すなわち、各冗長データは、係数行列の値が「1」の成分と積算された元データの EOR 演算結果となる。

40

【0105】

このような係数行列を要求された係数行列供給部 73 は、その要求に対して、後述するように、線形合同法を用いた擬似乱数により予め生成された係数行列を行列式生成部 72 に供給する。

【0106】

行列式生成部 72 は、係数行列を取得すると、その係数行列と 1 ブロック分の元データを用いて、そのブロックに対する冗長データを生成するための行列式を生成する。具体的には、行列式生成部 72 は、以下の式(1)に示されるような行列式を生成する。

【0107】

50

$$\begin{matrix} \text{【数1】} \\ \left(\begin{matrix} S_1 \\ S_2 \\ \vdots \\ S_M \end{matrix} \right) \end{matrix} = \begin{matrix} \left(\begin{matrix} 1 & 0 & 1 & \text{-----} & 0 \\ 0 & 1 & 0 & \text{-----} & 1 \\ \vdots & \vdots & \vdots & \diagdown & \vdots \\ 1 & 1 & 0 & \text{-----} & 1 \end{matrix} \right) \end{matrix} \times \begin{matrix} \left(\begin{matrix} D_1 \\ D_2 \\ \vdots \\ D_N \end{matrix} \right) \end{matrix} \quad \dots (1)$$

10

【0108】

式(1)において、右辺第1項は、M行N列の係数行列であり、係数行列供給部73より供給された係数行列である。右辺第2項は、元データD(D₁乃至D_N)の行列であり、左辺の行列は、この行列式により算出される冗長データS(S₁乃至S_M)である。なお、式(1)において、加算の代わりにビット毎のEOR演算が行われる。係数行列の各成分の値は、後述するように、ランダムに決定される。すなわち、式(1)は、各冗長データS₁乃至S_Mが係数行列により、元データD₁乃至D_Nの中から任意に選択された元データのEOR演算結果であることを示している。

【0109】

このような行列式を生成すると、行列式生成部72は、この行列式を行列式演算部74に供給する。なお、行列式は、所謂、連立方程式と同じものを示すものとする。以下においても同様である。

20

【0110】

行列式演算部74は、式(1)に示されるような行列式を演算する。すなわち、行列式演算部74は、右辺第1項の係数行列の各行において、その行の各列成分を、それらが対応する、右辺第2項における各行成分(元データD₁乃至D_N)とそれぞれ積算し、得られた各積算結果を用いて、互いに対応するビット毎にEOR演算を行うことにより、各行に1つのEOR演算結果を算出し、それを冗長データSとする。

【0111】

ここで、通常の10進数の場合の、2つの行列の積算における演算手順を説明すると、まず、第1項の行列の各行が第2項の行列の各列と積算される。具体的には、第1項の行列の各行と第2項の行列の各列が互いに対応する成分毎に積算される。例えば、第1項の第1行目について説明すると、その行の第i列目の成分は、第2項の各列の第i行目の成分と積算される。このように積算が行われると、次に、この積算毎に各成分の積算結果が全て加算される。例えば、第1項の行列がN列の行列であり、第2項の行列がN行の行列であるとすると、第1項の行列の各行と第2項の行列の各列の各積算において、それぞれ、N個の成分同士の積算結果が得られるので、それらのN個の積算結果が加算され、この加算結果が、第1項の行列の行と第2項の行列の列の積算結果とされる。

30

【0112】

上述した式(1)の行列式においても、基本的にはこのような演算手順と同様の演算手順により演算が行われる。しかしながら、元データDや冗長データSは、2進数のビット列であるので、上述した10進数の場合の演算手順における、成分同士の積算結果の加算処理の代わりに、ビット毎のEOR演算が用いられる。

40

【0113】

演算手順の順序で説明すると、行列式演算部74は、上述した10進数の場合と同様に、式(1)の右辺第1項の係数行列の各行を、右辺第2項の行列(元データD₁乃至D_Nを成分とする行列)の列と積算する。具体的には、右辺第1項の係数行列の各行の各列成分を、右辺第2項の行列の各成分(元データD₁乃至D_N)と、それぞれ積算する。右辺第1項の係数行列の各成分の値は「1」または「0」であるので、各成分の積算結果は、右辺第2項の各成分(元データD₁乃至D_N)となるか、若しくは、「0」(具体的には、元デ

50

ータDと同じビット数であり、各ビットの値が「0」のビット列)となる。

【0114】

次に、行列式演算部74は、これらの各成分の積算結果を、上述したように互いに対応するビット毎のEOR演算により合成する。なお、EOR演算について具体的に説明すると、値「0」と値「0」とのEOR演算結果の値は「0」であり、値「0」と値「1」とのEOR演算結果の値は「1」であり、値「1」と値「0」とのEOR演算結果の値は「1」であり、値「1」と値「1」とのEOR演算結果の値は「0」である。従って、例えば、「11110000」のビット列と、「10101010」のビット列とのEOR演算結果は、「01011010」となる。行列式演算部74は、このようなEOR演算を繰り返し、各成分の積算結果を全て合成する。そして、このEOR演算結果が、1つの冗長データSとされる。

10

【0115】

以上のような演算を、式(1)の右辺第1項の係数行列の各行に対して繰り返し、全ての左辺第1項の行列の成分として示される冗長データ S_1 乃至 S_M が求められる。

【0116】

なお、上述したように、係数行列の各成分の値は、「0」または「1」であるので、その各成分と元データ D_1 乃至 D_N との積算結果は、右辺第2項の各成分(元データ D_1 乃至 D_N)となるか、若しくは、「0」となる。従って、この積算について換言すると、行列式演算部62は、その各成分と元データ D_1 乃至 D_N との積算により、冗長データSを生成するために用いる元データDを選択(抽出)している。

【0117】

20

また、上述したように、各成分の積算結果が互いに対応するビット毎のEOR演算により合成されるので、合成結果(EOR演算結果)の各ビットの値は、そのビットが対応する、EOR演算される各積算結果のビットにおいて、値が「1」である積算結果の数が奇数である場合、「1」になり、値が「1」である積算結果の数が偶数である場合、「0」になる。従って、このEOR演算について換言すると、行列式演算部62は、上述した成分毎の積算結果の各ビットの値を、互いに対応するビット毎に集計し、値が「1」である積算結果の数が奇数である場合、そのビットに対応する冗長データSのビットの値を「1」とし、値が「1」である積算結果の数が偶数である場合、そのビットに対応する冗長データSのビットの値を「0」とする。

【0118】

30

すなわち、行列式演算部62は、元データ D_1 乃至 D_N の中から、所定の方法により元データDを選択し、その選択した各元データDの、互いに対応するビットについて、値が「1」の数が奇数である場合、そのビットに対応する冗長データSのビットの値を「1」とし、値が「1」の数が偶数である場合、そのビットに対応する冗長データSのビットの値を「0」として、各冗長データSを求める。

【0119】

行列式演算部62は、以上のように演算を行い、冗長データS(S_1 乃至 S_M)を求め、その解を解出力部75に供給する。すなわち、送信装置11においては、送信データより、図6に示されるようなパケットデータが生成されて送信される。

【0120】

40

図6において、元データ81は、パケットデータ81-1乃至81-Nにより構成され、冗長データ82は、パケットデータ82-1乃至82-Mにより構成される。元データ81であるパケットデータ81-1乃至81-Nは、それぞれ、元データ D_1 乃至 D_N に対応しており、冗長データ82であるパケットデータ82-1乃至82-Mは、元データ D_1 乃至 D_N の中から任意に選択された元データのEOR演算結果(排他的論理和)である。

【0121】

解出力部75は、それを、冗長データSとして、所定のタイミングで図2のアプリケーションデータ生成部34に供給する。

【0122】

次に、以上のようなEOR演算に用いられる係数行列について説明する。

50

【 0 1 2 3 】

図 7 は、図 5 の係数行列供給部 7 3 の詳細な構成例を示すブロック図である。

【 0 1 2 4 】

図 7 において、種データ保持部 9 1 は、RAM等の記憶素子により構成され、後述する擬似乱数の生成に用いられる種データを保持する。なお、種データとしてはどのような値であってもよいが、以下においては、予め定められた任意の定数を種データとする場合について説明する。

【 0 1 2 5 】

擬似乱数生成部 9 2 は、例えば、電源投入時等の所定のタイミングになると、種データ保持部 9 1 に種データを要求する（図中点線矢印）。種データ保持部 9 1 は、その要求に基づいて、種データを擬似乱数生成部 9 2 に供給する。擬似乱数生成部 9 2 は、種データを取得すると、その種データに基づいて、以下の式（ 2 ）に示されるような線形合同法のモジュロ演算を行い、擬似乱数を生成する。

【 0 1 2 6 】

【 数 2 】

$$\text{rand} = 1664525L * \text{rand} + 1013904223L \quad \dots (2)$$

ただし、初期値 $\text{rand} = 0$

【 0 1 2 7 】

式（ 2 ）において、変数 L は、種データである。また、右辺の変数 rand は、1 回前の演算により算出された擬似変数（すなわち、1 回前の演算における左辺の rand）である。変数 rand の初期値は「 0 」である。擬似乱数生成部 9 2 は、式（ 2 ）のモジュロ演算を例えば 3 2 ビット処理で行う。

【 0 1 2 8 】

擬似乱数生成部 9 2 は、式（ 2 ）を演算し、変数 rand（擬似乱数）を算出すると、その値を擬似乱数バッファ 9 3 に供給して保持させるとともに、係数判定部 9 4 に供給する。擬似乱数バッファ 9 3 に保持されている変数 rand は、擬似乱数生成部 9 2 に再度読み出され、次の擬似乱数演算において、式（ 2 ）の右辺の変数 rand に代入され、利用される。

【 0 1 2 9 】

以上の処理を繰り返し、擬似乱数生成部 9 2 は、擬似乱数を $M \times N$ 個生成し、それらを順次、係数判定部 9 4 に供給する。

【 0 1 3 0 】

係数判定部 9 4 は、取得した擬似乱数の値に基づいて、係数行列の各成分の値を 1 つずつ判定する。このとき、係数判定部 9 4 は、取得した擬似乱数の値に対して、例えば、閾値を設け、取得した擬似乱数の値が、その閾値以上である場合は、係数行列の成分の値を「 1 」と判定し、閾値より小さい場合は、係数行列の成分の値を「 0 」と判定し、その判定結果を係数行列保持部 9 5 に供給し、保持させる。

【 0 1 3 1 】

その際、係数判定部 9 4 は、係数生成確率保持部 9 5 に保持されている係数生成確率 p を取得し、この係数生成確率 p の値に基づいて、閾値の値を決定する。すなわち、係数判定部 9 4 は、値が「 0 」と「 1 」で構成される係数行列の全成分の内、値が「 1 」の成分の割合が係数生成確率 p の値になるように（近づくように）、閾値の値を設定する。

【 0 1 3 2 】

係数生成確率 p は、予め定められた、「 0 」より大きく、かつ、「 1 」より小さい値の定数であり、係数行列の全成分の内、値が「 1 」の成分の割合がどの程度になるかを確率的に指定する値である。すなわち、例えば、係数生成確率 p の値が「 0 . 5 」であった場合、係数判定部 9 4 は、生成された係数行列において、値が「 0 」の成分の数と、値が「 1 」の成分の数とがほぼ 1 対 1 となるように（近似するように）閾値を設定する。例えば、擬似乱数の値が、均一の確率で 3 2 ビット全体に分布すると仮定した場合、係数判定部 9 4 は、閾値の値を「 2147483648 」に設定する。

10

20

30

40

50

【 0 1 3 3 】

以上のようにして、係数行列保持部 9 6 には、係数判定部 9 4 より 1 つずつ判定結果（「0」または「1」）が供給され、係数行列保持部 9 6 は、それらの判定結果を成分とする M 行 × N 行の係数行列を生成し保持する。

【 0 1 3 4 】

そして、係数行列出力部 9 7 は、図 5 の行列式生成部 7 2 より係数行列を要求されると、係数行列保持部 9 6 に、保持している係数行列を要求して取得し、その係数行列を行列式生成部 7 2 に供給する。

【 0 1 3 5 】

以上のようにして、係数行列供給部 7 3 は、係数行列を作成し、要求に基づいて、その係数行列を供給する。

10

【 0 1 3 6 】

なお、係数行列を算出するタイミングは、係数行列を要求される以前であれば、どのタイミングであってももちろんよい。また、係数行列は、全ブロック間で共通であってもよいし、ブロック毎に新たに生成し、互いに独立した行列としてもよい。ただし、図 8 のように、曲線 1 0 1 で示される、ブロック間で係数行列を共通にした場合の方が、曲線 1 0 2 で示される、ブロック毎に新たに係数行列を生成した場合よりも、受信側における復元能力が高い。図 8 は、復元能力の関係を示すグラフであり、横軸はパケットロス数を示し、縦軸は、復元能力（復元不能頻度の逆数）を示す。

【 0 1 3 7 】

20

次に、アプリケーションデータについて説明する。

【 0 1 3 8 】

図 9 は、図 2 のアプリケーションデータ生成部 3 4 の詳細な構成例を示すブロック図である。

【 0 1 3 9 】

最初に、パケットデータが供給される前に、元データカウンタ 1 1 2、ブロックカウンタ 1 1 3、および、冗長データカウンタ 1 2 3 は、初期化される。すなわち、元データカウンタ 1 1 2 およびブロックカウンタ 1 1 3 のカウント値は、値「0」に設定され、冗長データカウンタ 1 2 3 のカウント値は、オフセット値供給部 1 2 2 より供給されるオフセット値に設定される。オフセット値供給部 1 2 2 は、例えば、1 ブロックにおける元データのデータ数等のような所定の値をオフセット値として保持しており、その値を冗長データカウンタに供給し、後述するように、冗長データの packets 番号が、元データの packets 番号と重複しないように設定する。

30

【 0 1 4 0 】

元データ受付部 1 1 1 は、ブロック分割部 3 2 より元データが供給されると、その元データがブロックの最初のパケットデータである場合、元データカウンタ 1 1 2 とブロックカウンタ 1 1 3 をカウントアップさせる。また、ブロック分割部 3 2 より供給された元データがブロックの最初のパケットデータでない場合、元データ受付部 1 1 1 は、元データカウンタ 1 1 2 のみをカウントアップさせる。元データカウンタ 1 1 2 およびブロックカウンタ 1 1 3 は、所定のタイミングで、そのカウント値をヘッダ付加部 1 1 4 に供給する。

40

【 0 1 4 1 】

また、元データ受付部 1 1 1 は、ブロックの最後のパケットデータを処理した後に、元データカウンタ 1 1 2 のカウント値を初期化する。

【 0 1 4 2 】

さらに、元データ受付部 1 1 1 は、各カウンタを調整すると、元データをヘッダ付加部 1 1 4 に供給する。ヘッダ付加部 1 1 4 は、元データ受付部 1 1 1 より供給された元データに対して、元データカウンタ 1 1 2 およびブロックカウンタ 1 1 3 より供給されたカウント値を、図 1 0 A に示されるように、ヘッダとして元データに付加し、アプリケーションデータを生成する。

50

【 0 1 4 3 】

図 1 0 A において、アプリケーションデータ 1 3 1 は、ヘッダとして付加されたブロック番号 1 3 2 および元データパケット番号 1 3 3、並びに、元データ 1 3 4 により構成される。ブロック番号 1 3 2 は、各ブロックを識別するための番号であり、各ブロックに対して互いに異なる値が割り当てられる。その値は、ブロックカウンタ 1 1 3 より供給されるカウント値である。元データパケット番号 1 3 3 は、各元データを識別するための番号であり、各元データに対して互いに異なる値が割り当てられる。その値は、元データカウンタ 1 1 2 より供給されるカウント値である。すなわち、図 1 1 の表 1 5 1 に示されるように、ブロック内において、ブロック番号 1 3 2 の値（図 1 1 の場合「0」）は、各アプリケーションデータにおいて共通であり、元データパケット番号 1 3 3 の値は、1 つずつ増加している（図 1 1 の場合「0」乃至「9 9 9」）。

10

【 0 1 4 4 】

このようなヘッダが元データ 1 3 4 に付加されて、アプリケーションデータ 1 3 1 が生成される。なお、ブロック番号 1 3 2 は、受信装置 1 3 が取得したパケットデータがどのブロックに属するデータであるか（どのブロックデータに対応するか）を特定するための情報であり、元データパケット番号 1 3 3 は、受信装置 1 3 がロスしたパケットを特定するための情報である。これらの情報をパケットデータに付加しアプリケーションデータとすることにより、受信装置 1 3 は、アプリケーション層の処理において、これらの情報を参照することができる。すなわち、例えば、IP パケット等にもパケット番号が付加されるが、これは再送制御等に用いられる情報であり、データの復元処理等に利用することはできず、上述した情報とは用途や性質が異なる情報である。

20

【 0 1 4 5 】

図 9 に戻り、冗長データ受付部 1 2 1 は、冗長データ生成部 3 3 より冗長データが供給されると、冗長データカウンタ 1 2 3 をカウントアップさせる。冗長データカウンタ 1 2 3 およびブロックカウンタ 1 1 3 は、所定のタイミングで、そのカウント値をヘッダ付加部 1 1 4 に供給する。

【 0 1 4 6 】

また、冗長データ受付部 1 2 1 は、ブロックの最後のパケットデータを処理した後に、冗長データカウンタ 1 2 3 のカウント値を初期化し、オフセット値供給部 1 2 2 よりオフセット値を冗長データカウンタ 1 2 3 に供給させる。

30

【 0 1 4 7 】

さらに、冗長データ受付部 1 2 1 は、冗長データカウンタ 1 2 3 を調整すると、冗長データをヘッダ付加部 1 1 4 に供給する。ヘッダ付加部 1 1 4 は、冗長データ受付部 1 2 1 より供給された冗長データに対して、冗長データカウンタ 1 2 3 およびブロックカウンタ 1 1 3 より供給されたカウント値を、図 1 0 B に示されるように、ヘッダとして冗長データに付加し、アプリケーションデータを生成する。

【 0 1 4 8 】

図 1 0 B において、アプリケーションデータ 1 4 1 は、ヘッダとして付加されたブロック番号 1 4 2 および冗長データパケット番号 1 4 3、並びに、冗長データ 1 4 4 により構成される。ブロック番号 1 4 2 の値は、ブロックカウンタ 1 1 3 より供給されるカウント値である。冗長データパケット番号 1 4 3 は、各冗長データを識別するための番号であり、各冗長データに対して異なる値が割り当てられる。その値は、冗長データカウンタ 1 2 3 より供給されるカウント値である。すなわち、図 1 1 の表 1 5 1 に示されるように、ブロック内において、ブロック番号 1 4 2 の値（図 1 1 の場合「0」）は、各アプリケーションデータにおいて共通であり、冗長データパケット番号 1 4 3 の値は、1 つずつ増加している（図 1 1 の場合「1 0 0 0」乃至「1 0 9 9」）。図 1 1 の場合、元データの 1 ブロック辺りの数が 1 0 0 0 個に設定されているので、オフセット値が「1 0 0 0」に設定されている。

40

【 0 1 4 9 】

このようなヘッダが冗長データ 1 4 4 に付加されて、アプリケーションデータ 1 4 1 が

50

生成される。なお、ブロック番号142は、受信装置13が取得したパケットデータがどのブロックに属するデータであるかを特定するための情報であり、冗長データパケット番号143は、受信装置13が取得したパケットを整列させるための情報である。これらの情報をパケットデータに付加しアプリケーションデータとすることにより、受信装置13は、アプリケーション層の処理において、これらの情報を参照することができる。すなわち、例えば、IPパケット等にもパケット番号が付加されるが、これは再送制御等に用いられる情報であり、データの復元処理等に利用することはできず、上述した情報とは用途や性質が異なる情報である。

【0150】

以上のように、アプリケーションデータには、元データを含むものと、冗長データを含むものが存在する。これらのアプリケーションデータは、上述したようにUDPやIP等に基づいてパケット化され、ネットワークを介して受信装置13に供給される。

10

【0151】

このように、アプリケーションデータ生成部34がブロック番号を元データや冗長データに付加することにより、受信装置13は、後述するように、受信したデータのブロックを識別することができ、ブロック単位で処理することができる。

【0152】

また、アプリケーションデータ生成部34が元データパケット番号を元データに付加することにより、受信装置13は、後述するように、そのパケット番号に基づいて、ロスした(受信できなかった)元データを特定することができる。

20

【0153】

さらに、アプリケーションデータ生成部34が冗長データパケット番号を冗長データに付加することにより、受信装置13は、後述するように、元データを復元する際に、冗長データを整列させることができるので、元データを復元するための行列式を容易かつ正確に生成することができる。

【0154】

さらに、アプリケーションデータ生成部34が元データパケット番号を元データに付加し、冗長データパケット番号を冗長データに付加することにより、受信装置13は、後述するように、元データを復元する際に、予め用意されたM行N列の係数行列より必要な部分を容易かつ正確に切り出すことができるので、元データを復元するための行列式を容易かつ正確に生成することができる。

30

【0155】

次に、データの受信側について説明する。

【0156】

図12は、図1の受信装置13の詳細な構成例を示すブロック図である。

【0157】

ネットワーク12を介して受信装置13に送信されたパケットは、ケーブル210を介してイーサネット(R)(Ethernet(R))処理部221に供給される。イーサネット(R)処理部221は、イーサネット(R)の規格に基づいて、データリンク層および物理層の処理を行い、供給されたパケットからIPパケットを抽出し、IP処理部222に供給する。

40

【0158】

IP処理部222は、IPプロトコルに基づいて、ネットワーク層の処理を行い、供給されたIPパケットよりUDPパケットを抽出し、アプリケーションデータ抽出部223に供給する。アプリケーションデータ抽出部223は、UDPプロトコルに基づいてトランスポート層の処理を行い、UDPパケットより、アプリケーションデータを抽出し、それをバッファ224に供給する。

【0159】

バッファ224は、アプリケーションデータを保持し、アプリケーションデータに付加されたブロック番号に基づいて、アプリケーションデータをブロック単位で管理し、所定のタイミングでそれらをブロック毎にデータ分離部225に供給する。

50

【0160】

データ分離部225は、供給されたアプリケーションデータに付加されたパケット番号に基づいて、アプリケーションデータが元データを含むか、または冗長データを含むかで分類し、元データを含むアプリケーションデータをパケットロス判定部226に供給するとともに、冗長データを含むアプリケーションデータをロスパケット復元部228に供給する。

【0161】

送信装置11より送信されたパケットは、受信装置13に向けて送信されるが、様々な原因により、受信装置13に受信されない場合がある。従って、図13に示されるように、受信装置13が、1つのブロックを構成するパケットの一部をロスすることもある。図13においては、送信装置11より供給される元データ241の3つのパケットデータ241A乃至241Cと、冗長データ242の1つのパケットデータ242Aをロスしている。このような場合、受信装置13は、元データ D_1 乃至 D_3 が不足しており、受信できた元データ241だけでは、ブロックデータを復元することができない。そこで、受信装置13は、このような場合には、受信できた冗長データ142を用いて、ロスしたパケットデータ241A乃至241Cを復元する。

【0162】

パケットロス判定部226は、供給された元データのパケット番号に基づいて、パケットをロスしたか否かを判定する。ロスしたと判定した場合、パケットロス判定部226は、元データを含むアプリケーションデータをブロックデータ復元部227に供給するとともに、その判定結果と、元データを含むアプリケーションデータをロスパケット復元部228に供給する。パケットをロスしていない場合、パケットロス判定部226は、元データを含むアプリケーションデータをブロックデータ復元部227に供給する。

【0163】

ブロックデータ復元部227は、パケットロス判定部226より供給されたアプリケーションデータの元データを合成し、ブロックデータを復元する。また、受信装置13がパケットをロスし、元データが不足している場合は、ロスパケット復元部228において復元された元データを利用する。ブロックデータを復元したブロックデータ復元部227は、それをブロック合成部229に供給する。

【0164】

ロスパケット復元部228は、パケットロス判定部226より供給された、元データを含むアプリケーションデータと、データ分離部225より供給された、冗長データを含むアプリケーションデータを用いて、EOR演算を行い、行列式を解くことにより、ロスした元データを復元し、それをブロックデータ復元部227に供給する。

【0165】

ブロック合成部229は、ブロックデータ復元部227より供給されるブロックデータを一時的に保持し、出力するタイミングを調整することにより、ブロック同士を合成し、1つの連続する受信データ（すなわち、送信装置11における送信データ）として、受信装置13の外部に出力する。

【0166】

なお、上述したバッファ224以降のモジュールが実行する処理は、OSI階層モデルにおける、アプリケーション層、プレゼンテーション層、およびセッション層の処理である。

【0167】

このようにデータを受信し、ロスした元データをEOR演算により復元することにより、受信装置13は、高速であり、かつ、復元性能の高い復元処理を、製造コストを増大せずに実現し、より容易に、より正確にデータを受信することができる。

【0168】

次に、ロスパケットの復元について説明する。なお、以下においては、送信装置11より送信されたN個の元データおよびM個の冗長データSの内、K個の元データとL個の冗長データをロスする場合について説明する。

10

20

30

40

50

【 0 1 6 9 】

図 1 4 は、ロスパケット復元部 2 2 8 の詳細な構成例を示すブロック図である。

【 0 1 7 0 】

図 1 4 において、図 1 2 のパケットロス判定部 2 2 6 より供給される判定結果、および元データのアプリケーションデータは、判定結果取得部 2 5 1 に供給される。判定結果取得部は、それらの情報を取得すると、元データのアプリケーションデータを元データ保持部 2 5 2 に供給するとともに、判定結果に基づいて、行列式作成部 2 5 3 に行列式作成の指示を供給する。

【 0 1 7 1 】

また、図 1 2 のデータ分離部 2 2 5 より供給される冗長データのアプリケーションデータは、冗長データ保持部 2 5 4 に供給され、一時的に保持される。

10

【 0 1 7 2 】

判定結果取得部 2 5 1 より行列式の作成を指示された行列式作成部 2 5 3 は、元データ保持部 2 5 2 に元データを要求して取得し、冗長データ保持部 2 5 4 に冗長データを要求して取得し、さらに、係数行列供給部 2 5 5 に係数行列を要求して取得する。なお、係数行列供給部 2 5 5 は、図 7 の係数行列供給部 7 3 と基本的に同様の構成であり、同様に動作するので、その詳細な説明を省略する。

【 0 1 7 3 】

行列式作成部 2 5 3 は、以上のように各情報を取得すると、ロスした元データを演算する行列式に必要な演算用の冗長データ Q を算出するための行列式を生成する。ロスした元データは、後述するように、受信した元データと冗長データを用いて求めるが、受信した冗長データには、受信した元データの情報も含まれている場合があるため、第 1 段階として、まず、受信した元データの情報を受信した冗長データに代入し、ロスした元データの EOR 演算結果のみで構成される演算用冗長データ Q を生成する。行列式作成部 2 5 3 は、演算用冗長データ Q を生成するために、式 (3) を作成する。

20

【 0 1 7 4 】

【数 3】

$$\begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{M-L} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & \text{-----} & 1 \\ 0 & 1 & 1 & \text{-----} & 0 \\ \vdots & \vdots & \vdots & \diagdown & \vdots \\ 1 & 1 & 0 & \text{-----} & 0 \end{pmatrix} \times \begin{pmatrix} R_1 \\ R_2 \\ \vdots \\ R_{N-K} \end{pmatrix} + \begin{pmatrix} J_1 \\ J_2 \\ \vdots \\ J_{M-L} \end{pmatrix} \dots (3)$$

30

【 0 1 7 5 】

なお、式 (3) において、加算は EOR 演算を示す。式 (3) の左辺の行列の各成分 Q_1 乃至 Q_{M-L} は、それぞれ、右辺第 3 項の行列の各成分である、受信した冗長データ J_1 乃至 J_{M-L} に対応する演算用冗長データである。式 (3) の右辺第 2 項の行列の各成分 R_1 乃至 R_{N-K} は受信した元データを示す。右辺第 1 項の係数行列は、予め係数行列供給部 2 5 5 において生成された M 行 N 列の係数行列より切り出された M - L 行 N - K 列の係数行列である。

40

【 0 1 7 6 】

行列式作成部 2 5 3 は、受信した冗長データ J_1 乃至 J_{M-L} を冗長データパケット番号に基づいて整列し、受信した元データ R_1 乃至 R_{N-K} を元データパケット番号に基づいて整列し、さらに、係数行列供給部 2 5 5 が保持している M 行 N 列の係数行列より、受信した元データ R_1 乃至 R_{N-K} および受信した冗長データ J_1 乃至 J_{M-L} に対応する部分の成分を切り出して、式 (3) を作成する。

【 0 1 7 7 】

50

式(3)に示されるように、演算用冗長データ Q_1 乃至 Q_{M-L} は、それぞれ、受信した冗長データ J_1 乃至 J_{M-L} から、受信した元データ R_1 乃至 R_{N-K} のEOR演算結果を除去したものである。なお、各冗長データは元データのEOR演算結果であるので、式(3)において、受信した冗長データ J は、再度、受信した元データ R をEOR演算することにより、冗長データ J に含まれる受信した元データの情報を除去することができる。

【0178】

すなわち、以上のようにして求められた演算用冗長データ Q_1 乃至 Q_{M-L} は、ロスした元データ P_1 乃至 P_K のEOR演算結果となる。

【0179】

行列式作成部253は、以上のような行列式を作成すると、それを行列式演算部256に供給する。行列式演算部256は、この行列式を演算し、演算用冗長データ Q を算出し、それを行列式作成部253に戻す。行列式作成部253は、演算用冗長データ Q を取得すると、今度は、それをを用いて元データを復元するための行列式を作成する。行列式作成部253は、受信した冗長データの packets 番号と、ロスした元データの packets 番号に基づいて、係数行列供給部255に保持されている M 行 N 列の係数行列より、それらの packets 番号に対応する部分($M-L$ 行 K 列)を切り出し、式(4)に示されるような行列式を生成する。

【0180】

【数4】

$$\begin{pmatrix} 0 & 0 & 1 & \cdots & 1 \\ 0 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \diagdown & \vdots \\ 1 & 0 & 0 & \cdots & 1 \end{pmatrix} \times \begin{pmatrix} P_1 \\ P_2 \\ \vdots \\ P_K \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{M-L} \end{pmatrix} \quad \dots (4)$$

【0181】

式(4)において、左辺第1項は、切り出した係数行列であり、左辺第2項の行列の各成分は、ロスした元データ P_1 乃至 P_K であり、右辺の行列の各成分は、演算用冗長データ Q_1 乃至 Q_{M-L} である。

【0182】

なお、演算用冗長データ Q_1 乃至 Q_{M-L} の数を示す($M-L$)の値は、ロスした元データの数を示す K の値より大きい必要がある。すなわち、式(4)において係数行列は、列数よりも行数の方が多い必要がある。換言すると、受信装置13は、すべての、ロスした元データを復元するためには、少なくとも、そのロスした元データの数よりも多く、冗長データを受信する必要がある。

【0183】

行列式作成部253は、式(4)に示されるような行列式を生成すると、それを行列式演算部256に供給する。行列式演算部256は、この行列式を取得すると、例えば、ガウスの消去法を用いて、ロスした元データ P_1 乃至 P_K の解を求める。求解方法の例については後述するが、ロスした元データを算出することができるものであれば、どのようなものであってもよい。

【0184】

ロスした元データ P_1 乃至 P_K の解を求めると、行列式演算部256は、その解を解出力部257に供給する。解出力部257は、その解よりロスした元データのアプリケーションデータを復元し、それを図12のブロックデータ復元部227に供給する。

【0185】

以上のように受信処理を行うことにより、受信装置13は、送信装置11に再送要求を

10

20

30

40

50

することなしに、ロスした元データを復元することができる。また、この復元処理のための冗長データをEOR演算により生成するため、演算の付加が軽減し、容易に冗長データを生成したり、冗長データを用いて復元処理を行ったりすることができる。従って、高速であり、かつ、復元性能の高い復元処理を、送信装置11および受信装置13の製造コストを増大せず実現し、より容易に、より正確な通信を行うことができる

【0186】

次に、送信装置11における、上述した各部の処理の流れについて説明する。

【0187】

外部からの送信データの供給が開始されると、送信装置11は、データ送信処理を開始する。図2の送信装置11によるデータ送信処理を図15のフローチャートを参照して説明する。

10

【0188】

最初に、送信データが外部より供給されると、送信装置11のデータ分割部31は、ステップS1において、送信データをブロックに分割し、分割したブロックデータをブロック分割部32に供給する。ブロック分割部32は、ステップS2において、供給されたブロックデータをさらにパケットデータに分割し、それを元データとして、冗長データ生成部33およびアプリケーションデータ生成部34に供給する。なお、以下の処理は、ブロック単位で行われる。

【0189】

パケットデータを供給された冗長データ生成部33は、ステップS3において、冗長データ生成処理を実行して、EOR演算による冗長データを生成し、それをアプリケーションデータ生成部34に供給する。冗長データ生成処理の詳細については、図16のフローチャートを参照して説明する。

20

【0190】

元データおよび冗長データを供給されたアプリケーションデータ生成部34は、ステップS4において、アプリケーションデータ生成処理を実行し、アプリケーションデータを生成し、生成したアプリケーションデータをUDPパケット化部35に供給する。なお、アプリケーションデータ生成処理の詳細については、図18のフローチャートを参照して後述する。

【0191】

アプリケーションデータを供給されたUDPパケット化部35は、ステップS5において、アプリケーションデータをUDPパケット化し、それをバッファ36に供給する。バッファ36は、供給されたUDPパケットを一時保持し、出力制御部37がステップS6において、そのバッファ36に保持されているUDPパケットを任意の順序で取り出すことで、UDPパケットの送信順を変更する。出力制御部37は、取り出したUDPパケットをIP処理部38に供給する。ステップS7において、IP処理部38は、供給されたUDPパケットをIPパケット化してイーサネット(R)処理部39に供給し、イーサネット(R)処理部39は、供給されたIPパケットをさらにパケット化するなどして、出力制御部37が変更した順番で、そのパケット(それぞれUDPパケットを含む)を、ケーブル40を介してネットワーク12(受信装置13)に送信する。

30

40

【0192】

そして、ブロック分割部32は、ステップS8において、外部より送信装置11に供給された送信データを構成する全ブロックに対して処理が終了したか否かを判定し、終了していないと判定した場合は、ステップS2に処理を戻し、それ以降の処理を繰り返す。また、全ブロックに対して送信処理が終了していないと判定した場合、ブロック分割部32は、ステップS2に処理を戻し、次のブロックに対してそれ以降の処理を繰り返す。

【0193】

以上のようにしてデータ送信処理が実行される。これにより、送信装置11は、ガロア体の演算など、負荷の大きい処理を行わずに、高速に冗長データを生成することができ、データを容易かつ正確に送信することができる。また、送信装置11は、パケットデータ

50

の送信順をシャッフルし、パケット番号と独立した順序で送信するので、バースト的パケットロスへの耐性も上げることができる。さらに、このように送信装置 11 が送信処理を行うことにより、送信装置 11 が送信したパケットを受信する受信装置 13 は、高速であり、かつ、復元性能の高い復元処理を、製造コストを増大せず実現することができる。すなわち、通信システム 1 は、より容易に、より正確な通信を行うことができる。

【0194】

次に、図 16 のフローチャートを参照して、図 15 のステップ S3 において実行される冗長データ生成処理について説明する。

【0195】

ブロック分割部 32 より元データを供給された冗長データ生成部 33 の元データ保持部 71 は、ステップ S21 において、供給された元データを一時的に保持していき、1 ブロック分保持する。

10

【0196】

元データ保持部 71 が 1 ブロック分の元データを保持すると、行列式生成部 72 は、ステップ S23 において、係数行列供給部 73 に対して係数行列を要求する。係数行列供給部 73 は、その要求に基づいて、予め用意した係数行列を行列式生成部 72 に供給する。係数行列の準備処理については、図 17 のフローチャートを参照して後述する。

【0197】

係数行列を取得した行列式生成部 72 は、ステップ S23 において、元データ保持部 71 に保持されている 1 ブロック分の元データを取得して、式 (1) のような、任意の元データの EOR 演算により冗長データを生成するための行列式を生成する。行列式を生成した行列式生成部 72 は、それを行列式演算部 74 に供給する。

20

【0198】

行列式を供給された行列式演算部 74 は、ステップ S24 において、EOR 演算を行い、その行列式を演算して、その解を解出力部 75 に供給する。

【0199】

解出力部 75 は、ステップ S25 において、供給された解を、ロスパケット復元処理用の冗長データとして、アプリケーションデータ生成部 34 に出力し、処理を図 15 のステップ S4 に戻してそれ以降の処理を実行させる。

【0200】

30

以上のように、元データを用いて EOR 演算を行い、冗長データを生成するので、冗長データ生成部 33 は、容易、かつ、高速に冗長データを生成することができる。これにより送信装置 11 は、ガロア体の演算など、負荷の大きい処理を行わずに、高速に冗長データを生成することができ、データを容易かつ正確に送信することができる。また、このように送信装置 11 が送信処理を行うことにより、送信装置 11 が送信したパケットを受信する受信装置 13 は、高速であり、かつ、復元性能の高い復元処理を、製造コストを増大せず実現することができる。すなわち、通信システム 1 は、より容易に、より正確な通信を行うことができる。

【0201】

次に、図 17 のフローチャートを参照して、図 7 の係数行列供給部 73 により実行される係数行列準備処理について説明する。係数行列供給部 73 は、例えば、送信装置 11 の電源投入時等に、この係数行列準備処理を実行する。

40

【0202】

係数行列準備処理を開始した係数行列供給部 73 の係数判定部 94 は、最初に、ステップ S41 において、係数生成確率保持部 95 より係数生成確率 p を取得し、その係数生成確率 p に基づいて、係数の値を決定する閾値を設定する。

【0203】

次に、擬似乱数生成部 92 は、ステップ S42 において、種データ保持部 91 に保持されている種データを取得し、さらに、擬似乱数バッファ 93 に保持されている前回の演算結果である擬似乱数を取得し、ステップ S43 において、取得した種データに基づいて、

50

上述した式(2)のような線形合同法により擬似乱数を生成する。擬似乱数を生成した擬似乱数生成部92は、生成した擬似乱数を擬似乱数バッファ92に供給して保持させるとともに、生成した擬似乱数を係数判定部94にも供給する。

【0204】

係数判定部94は、ステップS44において、供給された擬似乱数の値が、ステップS41において設定した閾値以上であるか否かを判定し、閾値以上であると判定した場合、ステップS45に処理を進め、今回出力する係数行列の成分の値を「1」に設定し、その成分を係数行列保持部96に出力し、ステップS47に処理を進める。逆に、ステップS44において、供給された擬似乱数の値が、閾値より小さいと判定した場合、係数判定部94は、ステップS46に処理を進め、今回出力する係数行列の成分の値を「0」に設定し、その成分を係数行列保持部96に出力し、ステップS47に処理を進める。

10

【0205】

係数行列の成分を供給された係数行列保持部96は、ステップS47において、その係数行列の成分を保持する。そして、係数行列保持部96は、ステップS48において、保持した成分により、係数行列が完成したか否かを判定し、完成していないと判定した場合、ステップS42に処理を戻し、それ以降の処理を各部に実行させ、次の成分を供給させる。

【0206】

ステップS48において、すべての成分が供給され、係数行列が完成したと判定した場合、係数行列保持部96は、係数行列準備処理を終了する。

20

【0207】

以上のように生成され、係数行列保持部96に保持されている係数行列は、行列式生成部72により要求された係数行列出力部97により、要求され、出力される。

【0208】

以上のように、擬似乱数を用いて係数行列の成分の値を決定することにより、冗長データ生成部33は、1ブロック分の元データからランダムに選択した元データを用いて、冗長データを生成することができ、冗長データの生成に用いられる元データの分布の偏りを抑制することができる。従って、受信装置13の復元処理において、元データ復元の可能性がロスした冗長データの内容に依存するのを抑制することができる。また、以上のように処理することにより、係数判定部94は、係数生成確率pによって冗長データに含まれる元データの数を確率的に制御することができる。

30

【0209】

次に、図15のステップS4において実行されるアプリケーションデータ生成処理の詳細を図18のフローチャートを参照して説明する。

【0210】

例えば、電源投入時等、パケットデータが供給される前に、ステップS61において、アプリケーションデータ生成部34の元データ受付部111は、元データカウンタ112およびブロックカウンタ113のカウンタ値をそれぞれ初期化し、例えば値「0」に設定する。また、アプリケーションデータ生成部34の冗長データ受付部121も、冗長データカウンタ123のカウンタ値を初期化し、オフセット供給部122より供給されるオフセット値に設定する。

40

【0211】

ステップS62において、元データ受付部111は、元データを受け付けたか否かを判定し、ブロック分割部32より供給された元データを受け付けたと判定した場合、ステップS63に処理を進め、受け付けた元データがブロック内の最初のパケットデータであるか否かを判定する。ブロック内の最初のパケットデータであると判定した場合、元データ受付部111は、ステップS64に処理を進め、ブロックカウンタ113をカウンタアップさせてカウンタ値を「1」増加させ、処理をステップS65に進める。なお、ステップS63において、既に、そのブロックの元データを受け付けており、受け付けた元データがブロック内の最初のパケットデータでないと判定した場合、元データ受付部111は、

50

ステップS 6 4の処理を省略し、ステップS 6 5に処理を進める。

【0 2 1 2】

ステップS 6 5において、元データ受付部1 1 1は、元データカウンタ1 1 2をカウントアップさせてカウント値を「1」増加させるとともに、受け付けた元データをヘッダ付加部1 1 4に供給する。元データを供給されたヘッダ付加部1 1 4は、ステップS 6 6において、元データカウンタ1 1 2およびブロックカウンタ1 1 3よりそれぞれカウント値を取得し、元データにブロックカウンタ1 1 3のカウント値をブロック番号として、元データカウンタ1 1 2のカウント値を元データパケット番号として、それぞれ付加し、ステップS 6 7において、カウント値を付加した元データをアプリケーションデータとしてUDPパケット化部3 5に供給する。

10

【0 2 1 3】

ステップS 6 7の処理を終了したヘッダ付加部1 1 4は、ステップS 6 8に処理を進める。また、ステップS 6 2において、元データを受け付けていないと判定した場合、元データ受付部1 1 1は、ステップS 6 3乃至ステップS 6 7の処理を省略し、ステップS 6 8に処理を進める。

【0 2 1 4】

ステップS 6 8において、冗長データ受付部1 2 1は、冗長データを受け付けたか否かを判定し、冗長データ生成部3 3より供給された冗長データを受け付けたと判定した場合、ステップS 6 9に処理を進め、冗長データカウンタ1 2 3をカウントアップさせてカウント値を「1」増加させるとともに、受け付けた冗長データをヘッダ付加部1 1 4に供給する。冗長データを供給されたヘッダ付加部1 1 4は、ステップS 7 0において、ブロックカウンタ1 1 3および冗長データカウンタ1 2 3よりそれぞれカウント値を取得し、冗長データにブロックカウンタのカウント値をブロック番号として、冗長データカウンタのカウント値を冗長データパケット番号として、それぞれ付加し、ステップS 7 1において、カウント値を付加した冗長データをアプリケーションデータとしてUDPパケット化部3 5に供給する。

20

【0 2 1 5】

ステップS 7 1の処理を終了したヘッダ付加部1 1 4は、ステップS 7 2に処理を進める。また、ステップS 6 8において、冗長データを受け付けていないと判定した場合、冗長データ受付部1 2 1は、ステップS 6 9乃至ステップS 7 1の処理を省略し、ステップS 7 2に処理を進める。

30

【0 2 1 6】

ステップS 7 2において、ヘッダ付加部1 1 4は、ブロックの元データおよび冗長データを全て処理したか否かを判定し、まだ未処理のパケットデータが存在すると判定した場合、ステップS 6 2に処理を戻し、それ以降の処理を繰り返す。ステップS 7 2において、全てのパケットデータを処理したと判定した場合、ヘッダ付加部1 1 4は、アプリケーションデータ生成処理を終了し、図1 5のステップS 5に処理を戻す。

【0 2 1 7】

以上のように、アプリケーションデータ生成部3 4のヘッダ付加部1 1 4がブロック番号を元データや冗長データに付加することにより、受信装置1 3は、後述するように、受信したデータのブロックを識別することができ、ブロック単位で処理することができる。

40

【0 2 1 8】

また、ヘッダ付加部1 1 4が元データパケット番号を元データに付加することにより、受信装置1 3は、後述するように、そのパケット番号に基づいて、ロスした（受信できなかった）元データを特定することができる。

【0 2 1 9】

さらに、ヘッダ付加部1 1 4が冗長データパケット番号を冗長データに付加することにより、受信装置1 3は、後述するように、元データを復元する際に、冗長データを整列させることができるので、元データを復元するための行列式を容易かつ正確に生成することができる。

50

【0220】

さらに、ヘッダ付加部114が元データパケット番号を元データに付加し、冗長データパケット番号を冗長データに付加することにより、受信装置13は、後述するように、元データを復元する際に、予め用意されたM行N列の係数行列より必要な部分を容易かつ正確に切り出すことができるので、元データを復元するための行列式を容易かつ正確に生成することができる。

【0221】

次に、受信装置13が実行する処理の流れについて説明する。

【0222】

最初に、図12の受信装置13によるデータ受信処理を図19のフローチャートを参照して説明する。 10

【0223】

受信装置13に対して、送信装置11よりパケットが送信され、そのパケットがケーブル210を介してイーサネット(R)処理部221に供給されると、イーサネット(R)処理部221は、ステップS91において、そのパケットを受信する。パケットを受信したイーサネット(R)処理部221は、そのパケットからIPパケットを抽出し、それをIP処理部222に供給する。IP処理部222は、そのIPパケットからUDPパケットを抽出し、それをアプリケーションデータ抽出部223に供給する。

【0224】

UDPパケットを供給されたアプリケーションデータ抽出部223は、ステップS92において、そのUDPパケットからアプリケーションデータを抽出し、それをバッファ224に供給する。バッファ224は、ステップS93において、その供給されたアプリケーションデータを保持し、ステップS94において、所定の時間が経過したか否かを判定することにより、1ブロック分のアプリケーションデータを保持したか否かを判定する。すなわち、バッファ224は、各ブロックのアプリケーションデータを、所定の時間蓄積していき、所定の時間が経過すると、それまでに蓄積した対象ブロックのアプリケーションデータをデータ分離部225に供給する。 20

【0225】

ステップS94において、所定の時間が経過しておらず、まだ1ブロック分のアプリケーションデータを保持していないと判定した場合、バッファ224は、ステップS91に処理を戻し、それ以降の処理を繰り返す。すなわち、受信装置13は、ステップS91乃至ステップS94の処理を繰り返しながら、バッファ224に所定の時間が経過するまで待機する。 30

【0226】

ステップS94において、所定の時間が経過したことにより、1ブロック分のアプリケーションデータを保持したと判定した場合、バッファ224は、保持しているアプリケーションデータ(所定の時間の間に保持した、対象ブロックのアプリケーションデータ)を、データ分離部225に供給し、処理をステップS95に進める。ステップS95において、データ分離部225は、供給されたアプリケーションデータを、そのパケット番号等に基づいて、元データと冗長データに分離し、元データのアプリケーションデータをパケットロス判定部226に供給するとともに、冗長データのアプリケーションデータをロスパケット復元部228に供給する。 40

【0227】

アプリケーションデータを供給されたパケットロス判定部226は、ステップS96において、供給されたアプリケーションデータの元データパケット番号に基づいて、ロスしたパケットを確認し、ステップS97において、元データをロスしたか否かを判定する。元データをロスしたと判定した場合、パケットロス判定部226は、元データのアプリケーションデータをブロックデータ復元部227に供給するとともに、その判定結果と、元データのアプリケーションデータをロスパケット復元部228に供給し、処理をステップS98に進める。ロスパケット復元部228は、ステップS98において、元データ復元 50

処理を実行してロスした元データを復元し、復元した元データのアプリケーションデータをブロックデータ復元部 227 に供給する。なお、元データ復元処理の詳細については、図 20 のフローチャートを参照して後述する。

【0228】

元データを復元したロス packets 復元部 228 は、処理をステップ S99 に進める。また、ステップ S97 において、全ての元データ packets 番号が揃っており、元データをロスしていないと判定した場合、元データのアプリケーションデータをブロックデータ復元部 227 に供給し、ステップ S98 の処理を省略して、ステップ S99 に処理を進める。

【0229】

ステップ S99 においてブロックデータ復元部 227 は、供給されたアプリケーションデータより元データを抽出し、それらを合体してブロックデータを復元し、復元したブロックデータをブロック合成部 229 に供給する。ブロック合成部 229 は、供給されたブロックデータを、一時的に保持し、所定のタイミングで出力することにより、ブロックデータ復元部 227 より順次供給されるブロックデータを連続して出力するようにしてブロックデータを合成し、送信前の送信データと同等の受信データとして受信装置 13 の外部に出力する。

【0230】

ブロック合成部 229 は、ステップ S100 において、データ受信処理を終了するか否かを判定し、1つの受信データとしての全てのブロックデータを処理しておらず、データ受信処理を終了しないと判定した場合、処理をステップ S91 に戻し、それ以降の処理を各部に繰り返させる。

【0231】

また、ステップ S100 において、データ受信処理を終了すると判定した場合、ブロック合成部 229 は、各部に終了処理を実行させ、データ受信処理を終了する。

【0232】

以上のようにしてデータ受信処理を実行することにより、受信装置 13 は、高速であり、かつ、復元性能の高い復元処理を、製造コストを増大せずに実現し、より容易に、かつ、より正確にデータを受信することができる。

【0233】

次に、図 19 のステップ S98 において実行される元データ復元処理の詳細を、図 20 のフローチャートを参照して説明する。

【0234】

ロス packets 復元部 228 の判定結果取得部 251 は、packets ロス判定部 226 より判定結果を取得すると、判定結果とともに供給される元データのアプリケーションデータを取得し、それを元データ保持部 252 に供給して保持させる。そして、判定結果取得部 251 は、行列式作成部 253 に行列式の作成を指示する。

【0235】

行列式作成部 253 は、その指示に基づいて、元データ復元処理を開始し、ステップ S121 において、元データ保持部 252 より元データのアプリケーションデータを取得し、ステップ S122 において、取得したアプリケーションデータの元データ packets 番号を参照して、ロスした元データの packets 番号を特定する。

【0236】

また、行列式作成部 253 は、ステップ S123 において、冗長データ保持部 254 に保持されている、データ分離部 225 より供給された冗長データのアプリケーションデータを取得し、ステップ S124 において、取得したアプリケーションデータの冗長データ packets 番号を参照し、その冗長データ packets 番号に基づいて、行列式を作成するために、冗長データを整列させる。

【0237】

さらに、行列式作成部 253 は、ステップ S125 において、取得したアプリケーションデータの元データ packets 番号および冗長データ packets 番号を参照し、それらのパケ

10

20

30

40

50

ット番号に基づいて、係数行列供給部 2 5 5 に保持されている係数行列 (M 行 N 列) より必要な部分 (M - L 行 N - K 列) を取得する。

【 0 2 3 8 】

なお、この処理は、係数行列供給部 2 5 5 が、係数行列から、行列式作成部 2 5 3 に要求された部分を切り出して、それを行列式作成部 2 5 3 に供給するようにしてもよいし、行列式作成部 2 5 3 が、係数行列供給部 2 5 5 より供給された M 行 N 列の係数行列から、必要な部分 (M - L 行 N - K 列) を切り出すようにしてもよい。

【 0 2 3 9 】

すなわち、行列式作成部 2 5 3 は、係数行列の、取得した冗長データに対応する行と、取得した元データに対応する列の、両方に対応する成分を抽出し、それらを成分とする新たな係数行列を取得する。

10

【 0 2 4 0 】

そして、ステップ S 1 2 6 において、行列式作成部 2 5 3 は、取得した係数行列および冗長データより、計算用冗長データを求める行列式 (例えば、式 (3)) を作成し、作成した行列式を行列式演算部 2 5 6 に供給する。

【 0 2 4 1 】

行列式演算部 2 5 6 は、ステップ S 1 2 7 において、供給された行列式を演算し、計算用冗長データを算出し、算出した計算用冗長データを行列式作成部 2 5 3 に戻す。

【 0 2 4 2 】

計算用冗長データを取得した行列式作成部 2 5 3 は、ステップ S 1 2 8 において、ロスした元データの packets 番号と、取得した冗長データの packets 番号に基づいて、係数行列供給部 2 5 5 に保持されている係数行列 (M 行 N 列) より必要な部分 (M - L 行 K 列) を取得する。

20

【 0 2 4 3 】

なお、この処理は、係数行列供給部 2 5 5 が、係数行列から、行列式作成部 2 5 3 に要求された部分を切り出して、それを行列式作成部 2 5 3 に供給するようにしてもよいし、行列式作成部 2 5 3 が、係数行列供給部 2 5 5 より供給された M 行 N 列の係数行列から、必要な部分 (M - L 行 K 列) を切り出すようにしてもよい。

【 0 2 4 4 】

すなわち、行列式作成部 2 5 3 は、係数行列の、取得した冗長データに対応する行と、ロスした元データに対応する列の、両方に対応する成分を抽出し、それらを成分とする新たな係数行列を取得する。

30

【 0 2 4 5 】

そして、ステップ S 1 2 9 において、行列式作成部 2 5 3 は、取得した係数行列および算出した計算用冗長データを用いて、ロスした元データを求めるための行列式 (例えば、式 (4)) を作成し、作成した行列式を行列式演算部 2 5 6 に供給する。

【 0 2 4 6 】

行列式演算部 2 5 6 は、ステップ S 1 3 0 において、求解処理を実行し、EOR演算によって供給された行列式の解を求め、その解を解出力部 2 5 7 に供給する。なお、求解処理の詳細については、図 2 1 および図 2 2 のフローチャートを参照して後述する。解出力部 2 5 7 は、ステップ S 1 3 1 において、供給された解を元データとしてアプリケーションデータを生成し、それをブロックデータ復元部 2 2 7 に供給し、元データ復元処理を終了して、図 1 9 のステップ S 9 9 に処理を戻す。

40

【 0 2 4 7 】

以上のように復元処理を行うことにより、ロス packets 復元部 2 2 8 は、高速であり、かつ、復元性能の高い復元処理を、製造コストを増大せずを実現することができる。従って、受信装置 1 3 は、より容易、かつ、より正確にデータを受信することができる。また、ロス packets 復元部 2 2 8 は、元データのアプリケーションデータに付加された元データ packets 番号を参照して、ロスした元データの packets 番号を特定するので、より容易に復元処理を行うことができる。さらに、ロス packets 復元部 2 2 8 は、冗長データの A

50

アプリケーションデータに付加された冗長データパケット番号を参照するので、容易に、受信した冗長データをパケット番号順に整列させることができ、予め用意された係数行列と、容易に対応付け、行列式を作成することができる。

【0248】

次に、図21および図22のフローチャートを参照して、図20のステップS130において実行される求解処理の詳細を説明する。

【0249】

ロスした元データを求めるための行列式を供給された行列式演算部256は、ステップS151において、行列式に含まれる係数行列(M-L行K列)を参照し、全ての列に値が「1」の成分が存在するか否かを判定する。係数行列の全列に、値が「1」の成分が少なくとも1つ存在しないと、その列に対応する、ロスした元データを算出することができないので、行列式演算部256は、ステップS151においてこの判定処理を行う。

10

【0250】

ステップS151において、全ての列に値が「1」の成分が少なくとも1つ存在すると判定した場合、行列式演算部256は、処理をステップS152に進め、変数iの値を「1」に設定し、ガウスの消去法の、前進消去を開始する。前進消去は、行列式の係数行列において、値が「1」の成分の数を減らすように、係数行列の行間でEOR演算を行い、行列式を変形していく処理である。この前進消去の処理の具体的な例については、後述する。

【0251】

前進消去処理を開始した行列式演算部256は、ステップS153に処理を進め、第i列の成分の値が「1」の行を、対象行の第1行目から順に検出する。対象行とは、係数行列において、前進消去処理の対象となる行のことである。後述するように、前進消去処理が進むごとに、この対象行が変化する。なお、初期状態において、対象行は全行である。

20

【0252】

ステップS154において、行列式演算部256は、ステップS153の処理により、行が複数検出されたか否かを判定し、複数検出された場合、ステップS155において、最初に検出された行と、2番目以降に検出された行とで各成分についてEOR演算を行う。このとき、行列式の他の項も、この係数行列のEOR演算に対応するようにEOR演算を行う。

30

【0253】

行同士のEOR演算を行った行列式演算部256は、ステップS156に処理を進める。また、ステップS154において、第i列の成分の値が「1」である行を、1行しか検出できなかったと判定した場合(各列には、値が「1」の成分が少なくとも1つ存在するので、1行は検出される)、ステップS155の処理を省略し、ステップS156に処理を進める。

【0254】

ステップS156において、行列式演算部256は、検出された(EOR演算により残された)、第i列の成分の値が「1」の行を、求解用に保持し、対象行から除外する。そして、行列式演算部256は、ステップS157に処理を進め、変数iの値に「1」を加算し、ステップS158において、変数iの値がKより大きいかが否かを判定する。変数iの値がK以下である(すなわち、未処理の列が存在する)と判定した場合、行列式演算部256は、処理をステップS153に戻し、次の列に対して、それ以降の処理を繰り返す。

40

【0255】

以上の処理を繰り返すことにより、行列式演算部256は、前進消去の処理を進め、全ての列に対して処理を行い、係数行列の値が「1」の成分の数を減らすようにする。

【0256】

そして、ステップS158において、変数iの値がKより大きいと判定した場合、行列式演算部256は、処理を図22のステップS161に進める。

【0257】

50

図 2 2 のステップ S 1 6 1 において、行列式演算部 2 5 6 は、変数 i の値を K に設定し、保持した行を用いて、後退代入を開始する。後退代入は、以上の前進消去により、値が「1」の成分が減らされた係数行列によって、元データの解を求め、求めた解を順次他の行の方程式に代入していく処理である。この後退代入の処理の具体的な例については、後述する。

【 0 2 5 8 】

後退代入を開始した行列式演算部 2 5 6 は、ステップ S 1 6 2 において、前進消去により i 番目に保持した行の方程式を解く。なお、後述するように、行列式演算部 2 5 6 は、それ以前に求められた解を方程式の変数（元データ）に代入して方程式を解く。

【 0 2 5 9 】

ステップ S 1 6 2 の処理を終了した、行列式演算部 2 5 6 は、ステップ S 1 6 3 に処理を進め、変数 i の値を「1」減算し、ステップ S 1 6 4 において、全ての解（元データ）を求めたか否かを判定する。未知の変数（元データ）が存在し、全ての解を求めていると判定した場合、行列式演算部 2 5 6 は、ステップ S 1 6 1 に処理を戻し、全ての解が求まるまで、それ以降の処理を繰り返す。

【 0 2 6 0 】

ステップ S 1 6 4 において、全ての解を求めたと判定した場合、行列式演算部 2 5 6 は、求解処理を終了し、図 2 0 のステップ S 1 3 1 に処理を戻す。

【 0 2 6 1 】

また、図 2 1 のステップ S 1 5 1 において、全ての列に値が「1」の成分が存在しないと判定した場合、上述したように、全ての解を求めることができないので、行列式演算部 2 5 6 は、図 2 2 のステップ S 1 6 5 に処理を進め、エラー処理を行い、ロスしたパッケージを復元できない旨のメッセージを出力する等して、求解処理を終了し、処理を図 2 0 のステップ S 1 3 1 に戻す。

【 0 2 6 2 】

以上のように行列式演算部 2 5 6 が、ガウスの消去法を用いて行列式を解くことにより、ロスパッケージ復元部 2 2 8 は、容易にロスした元データを復元することができる。これにより、受信装置 1 1 は、高速な復元処理を、製造コストを増大せずを実現し、より容易、かつより正確にデータを受信することができる。

【 0 2 6 3 】

次に、図 2 3 乃至図 2 6 を参照して、上述したガウスの消去法の具体的な例について説明する。なお、ここでは、説明の簡略化のため、5 個の演算用冗長データ X_1 乃至 X_5 を用いて、ロスした 4 個の元データ D_1 乃至 D_4 を求める場合について説明する。このとき、行列式の係数行列を、図 2 3 に示されるような成分の 5 行 \times 4 列の行列とし、行列式作成部 2 5 3 が図 2 3 に示されるような行列式を作成したとする。

【 0 2 6 4 】

図 2 3 に示される行列式を取得すると、行列式演算部 2 5 6 は、最初に、図 2 1 のステップ S 1 5 1 の処理を実行し、係数行列（図 2 3 において左辺第 1 項）の各列に値が「1」の成分が存在するか否かを判定する。図 2 3 の係数行列においては、全ての列に、値が「1」の成分が存在するので、元データ D_1 乃至 D_4 を求めることができる。

【 0 2 6 5 】

次に、行列式演算部 2 5 6 は、前進消去を開始し、1 回目の処理として、1 列目（ $i = 1$ ）を処理の対象とする。そして、行列式演算部 2 5 6 は、全ての行を対象行として、1 列目に注目し、値が「1」の成分が存在する行を対象行の 1 行目から順次検出する。図 2 3 の場合、点線枠 2 7 1 で示される 1 列目の成分の値が「1」である行が検出され、点線枠 2 8 1 乃至 2 8 3 で示される 1 行目、3 行目、および 5 行目がその順番で検出される。

【 0 2 6 6 】

1 列目の成分の値が「1」である行を検出した行列式演算部 2 5 6 は、次に、1 番目に検出した行と、2 番目以降に検出した行の各成分同士を EOR 演算する。図 2 3 の場合、行列式演算部 2 5 6 は、点線枠 2 8 1 で示される 1 行目と、点線枠 2 8 2 で示される 3 行目

10

20

30

40

50

とを成分ごとにEOR演算する。同様に、行列式演算部 2 5 6 は、点線枠 2 8 1 で示される 1 行目と、点線枠 2 8 3 で示される 5 行目とを成分ごとにEOR演算する。このEOR演算により、係数行列の各成分の値は、図 2 4 に示される係数行列（左辺第 1 項）のようになる。

【 0 2 6 7 】

また、このとき、行列式演算部 2 5 6 は、行列式が成立するように、右辺の行列においても係数行列に対応するようにEOR演算を行う。図 2 3 の場合、行列式演算部 2 5 6 は、右辺の演算用冗長データの行列に対して、点線枠 2 9 1 で示される 1 行目（X 1）と、点線枠 2 9 2 で示される 3 行目（X 3）とをEOR演算し、さらに、点線枠 2 9 1 で示される 1 行目（X 1）と、点線枠 2 9 3 で示される 5 行目（X 5）とをEOR演算する。このEOR演算により、演算用冗長データの行列の各成分の値は、図 2 4 に示される演算用冗長データの行列（右辺）のようになる。

10

【 0 2 6 8 】

以上のようにして、対象行における第 1 列目の成分の値が「 1 」である行を第 1 行目のみにすると、行列式演算部 2 5 6 は、次に、2 回目の処理に移行し、係数行列の第 2 列目を処理の対象とし（ $i = 2$ ）、第 1 行目は保持し、第 2 行目乃至第 5 行目を対象行とする。そして、行列式演算部 2 5 6 は、1 回目の場合と同様に、係数行列の第 2 列目について、値が「 1 」の成分が存在する行を対象行の 1 行目（図 2 5 の場合、係数行列の第 2 行目）から順次検出する。図 2 4 の場合、点線枠 3 0 1 で示される第 2 列目の成分の値が「 1 」である行が検出され、点線枠 3 1 1 および 3 1 2 で示される、係数行列の第 2 行目および第 5 行目がその順番で検出される。

20

【 0 2 6 9 】

第 2 列目の成分の値が「 1 」である行を検出した行列式演算部 2 5 6 は、1 回目の場合と同様に、1 番目に検出した行と、2 番目以降に検出した行の各成分同士をEOR演算する。図 2 4 の場合、行列式演算部 2 5 6 は、点線枠 3 1 1 で示される第 1 行目と、点線枠 3 1 2 で示される第 5 行目とを成分ごとにEOR演算する。このEOR演算により、係数行列の各成分の値は、図 2 5 に示される係数行列（左辺第 1 項）のようになる。

【 0 2 7 0 】

また、このとき、行列式演算部 2 5 6 は、1 回目の場合と同様に、行列式が成立するように、右辺の行列においても係数行列に対応するようにEOR演算を行う。図 2 4 の場合、行列式演算部 2 5 6 は、右辺の演算用冗長データの行列に対して、点線枠 3 2 1 で示される第 2 行目（X 2）と、点線枠 3 2 2 で示される第 5 行目（X 5 + X 1）とをEOR演算する。このEOR演算により、演算用冗長データの行列の各成分の値は、図 2 5 に示される演算用冗長データの行列（右辺）のようになる。

30

【 0 2 7 1 】

以上のようにして、対象行における第 2 列目の成分の値が「 1 」である行を第 2 行目のみにすると、行列式演算部 2 5 6 は、次に、3 回目の処理に移行し、係数行列の第 3 列目を処理の対象とし（ $i = 3$ ）、第 2 行目をさらに保持し、第 3 行目乃至第 5 行目を対象行とする。そして、行列式演算部 2 5 6 は、1 回目の場合と同様に、係数行列の第 3 列目について、値が「 1 」の成分が存在する行を対象行の 1 行目（図 2 5 の場合、係数行列の第 3 行目）から順次検出する。図 2 5 の場合、点線枠 3 3 1 で示される第 3 列目の成分の値が「 1 」である行が検出され、点線枠 3 4 1 で示される、係数行列の第 4 行目のみが検出される。

40

【 0 2 7 2 】

従って、この場合、行列式演算部 2 5 6 は、EOR演算を省略する。また、このとき、行列式演算部 2 5 6 は、EOR演算を省略したので、右辺の行列においてもEOR演算を省略する。図 2 5 の場合、右辺の演算用冗長データの行列において、点線枠 3 5 1 で示される第 4 行目（X 4）が係数行列の第 4 行目に対応するが、EOR演算は省略される。従って、この 3 回目の処理により行列式の各項の行列の各成分の値は、図 2 6 に示されるように変化せず、図 2 5 の場合と同様になる。

【 0 2 7 3 】

50

行列式演算部 2 5 6 は、次に、4 回目の処理に移行し、係数行列の第 4 列目を処理の対象とし ($i = 4$)、前回検出した第 4 行目をさらに保持し、第 3 行目および第 5 行目を対象行とする。そして、行列式演算部 2 5 6 は、1 回目の場合と同様に、係数行列の第 4 列目について、値が「1」の成分が存在する行を対象行の 1 行目 (図 2 5 の場合、係数行列の第 3 行目) から順次検出する。図 2 5 の場合、点線枠 3 6 1 で示される第 4 列目の成分の値が「1」である行が検出され、点線枠 3 7 1 で示される、係数行列の第 3 行目のみが検出される (第 5 行目第 4 列目の成分の値は「0」である)。

【0 2 7 4】

従って、この場合も、3 回目の処理と同様に、行列式演算部 2 5 6 は、EOR 演算を省略する。また、このとき、行列式演算部 2 5 6 は、EOR 演算を省略したので、右辺の行列において EOR 演算を省略する。図 2 6 の場合、右辺の演算用冗長データの行列において、点線枠 3 8 1 で示される第 3 行目 ($X_3 + X_1$) が係数行列の第 3 行目に対応するが、EOR 演算は省略される。従って、この 4 回目の処理によっても行列式の各項の値は変化しない。行列式演算部 2 5 6 は、この 4 回目の処理において検出された第 3 行目をさらに保持する。

10

【0 2 7 5】

以上のように、全列に対して前進消去を行うと行列式演算部 2 5 6 は、前進消去を終了し、次に後退代入を開始する。前進消去により、最後に保持した行には、値が「1」の成分が最後の列の 1 つしか存在しない。その他の列の成分の値は、それまでの EOR 演算により「0」に変換されている。同様に、最後から 2 番目に保持した行の、値が「1」の成分は 2 個以下であり、その前に保持した行は 3 個以下である。このように、最後から Y 番目に保持した行には、値が「1」の成分が Y 個以下となる。

20

【0 2 7 6】

従って、行列式演算部 2 5 6 は、後退代入により、最後に保持した行から解を 1 つ求め、保持した順番と逆順にその解を代入していき、全ての解を求める。

【0 2 7 7】

このように、ガウスの消去法を用いることにより、行列式演算部 2 5 6 は、EOR 演算等の単純な演算により行列式を解くことができるので、ロスパケット復元部 2 2 8 は、ロスした元データを容易にかつ高速に復元することができる。

【0 2 7 8】

次に、係数生成確率 p の値の決定方法について説明する。

30

【0 2 7 9】

図 2 7 および図 2 8 は、受信装置 1 3 におけるパケット復元能力を示すグラフであり、いずれの場合も、横軸がパケットのロス数を示し、縦軸が復元能力 (復元回数/復元不能の発生回数) を示している。図 2 7 は、係数生成確率 p の値が $1/2$ である場合の復元能力を示しており、図 2 8 は、係数生成確率 p の値が $1/6$ である場合の復元能力を示している。

【0 2 8 0】

図 2 7 と図 2 8 に示されるように、受信装置 1 3 の、ロスしたパケットの復元能力は、パケットロス数の減少するとともに増加し、係数生成確率 p が高いほど復元能力は高くなる。しかしながら、当然、係数生成確率 p が高いほど、復元処理における演算量が増加するので、復元処理の負荷が大きくなる。後述するように、例えば、リードソロモン符号を用いた復元処理と比較すると、EOR 演算処理による復元処理の負荷は非常に少ないが、係数生成確率 p の値が小さいほど、より高速に復元処理を行うことができる。従って、処理の負荷の観点からすれば、係数生成確率 p の値をなるべく小さく設定する方が望ましいが、上述したように、係数生成確率 p の値を大きくする方が復元能力は高くなる。

40

【0 2 8 1】

つまり、処理の負荷の大きさと復元能力の高さはトレードオフの関係にあり、係数生成確率 p の値は、実際のシステムにおいて、ネットワーク 1 2 におけるパケットロス率がどの程度であるかを考慮した上で、復元能力が実用的な値となる範囲において、なるべく小

50

さく設定するのが望ましい。

【0282】

以上のように、通信システム1において、送信装置11は、送信用データの他に、その送信用データよりEOR演算を行って生成した、パケットロスに対する誤り訂正用の冗長データを複数生成し、それらを送信用データとともに受信装置13に送信する。受信装置13は、その冗長データ(受信した冗長データ)に基づいて、受信できなかった元の送信用データの復元処理を行う。このようにすることにより、受信装置13は、送信装置11に再送要求を行うことなく、ロスしたデータを復元することができる。従って、通信システム1は、より容易に、より正確な通信を行うことができる。

【0283】

図29は、上述したEOR演算による復元処理(EOR)の処理時間と、リードソロモン符号を用いた復元処理(リードソロモンFEC(Forward Error Correction))の処理時間との関係の例を示す図である。

【0284】

図29においては、送信されるコンテンツのブロックサイズを5秒とし、コンテンツのビットレートを4Mbpsとし、パケットデータのデータサイズを1KByteとする。この場合、1ブロックは2500パケットとなる。図29の表391は、このような通信システム1において、パケットロスを10%(250パケット)としたときの、EOR演算による復元処理の処理時間と、リードソロモン符号を用いた復元処理の処理時間との関係を示している。

【0285】

表391に示されるように、EOR演算による復元処理の方が、リードソロモン符号を用いた復元処理と比較して、前処理、ガウスの消去による演算処理、データ復元処理、および、それらの総計の、全てにおいて非常に短く、また、CPU(Central Processing Unit)占有率(総計欄のカッコ内数字)も圧倒的に少ない。

【0286】

従って、受信装置13は、リードソロモン符号を用いた復元方法の場合と比較して、非常に高速に、または、処理の負荷をかけずに復元処理を行うことができる。

【0287】

図30は、上述したEOR演算による復元処理(EOR)の復元能力と、リードソロモン符号を用いた復元処理の復元能力(RS-FEC)との関係の例を示す図である。

【0288】

図30の表401においては、1ブロックのパケット数として実用的な値を設定しており、EOR演算による復元処理の場合、1000パケットおよび2000パケットの例を示し、リードソロモンFECの場合、100パケットの例を示す。

【0289】

表401において、エラー率(復元失敗率)が 10^{-8} となる時のパケットロス率と、エラー率が 10^{-14} となるパケットロス率が表示される。これらの値は、大きいほど、マージンがあることを示す。表401に示されるように、EOR演算を用いた復元処理の方が、リードソロモン符号を用いた復元処理の場合と比較してマージンが大きい。なお、エラー率 10^{-8} は、リードソロモンFECの場合では連続視聴で約一年に一回の割合であり、EOR演算による復元処理の場合では10年に一回の割合である。従って、図1の通信システム1は、より容易かつより正確に処理を実行することができる。

【0290】

なお、以上においては、アプリケーションデータ生成部34のヘッダ付加部114は、パケット番号を元データや冗長データに付加するように説明したが、アプリケーションデータのパケット番号の部分は、元データのパケット番号と、冗長データのパケットが識別可能な状態で供給されればよく、例えば、図31に示されるように、オフセット値の代わりに識別子が用いられるようにしてもよい。

【0291】

10

20

30

40

50

図 3 1 においては、アプリケーションデータ 4 1 1 は、データ（元データまたは冗長データ）4 1 4 に、ブロック番号 4 1 2 およびパケット番号 4 1 3 が付加されたデータであるが、パケット番号 4 1 3 の左 1 ビットは、元データ / 冗長データ識別フィールド 4 2 1 に設定され、残りのビット列がパケット番号フィールド 4 2 2 に設定される。

【 0 2 9 2 】

すなわち、この場合、冗長データパケット番号は、オフセット値が付加されておらず、元データ / 冗長データ識別フィールド 4 2 1 のデータ識別子により、元データパケット番号と識別される。例えば、データ識別子の値が「 0 」である場合、元データのアプリケーションデータであることを示し、そのパケット番号は元データパケット番号であることを示し、データ識別子の値が「 1 」である場合、冗長データのアプリケーションデータであることを示し、そのパケット番号は冗長データパケット番号であることを示している。

10

【 0 2 9 3 】

なお、このデータ識別子の値と、その値の示すデータの関係が上述した場合と逆であっても良い。また、元データ / 冗長データ識別フィールド 4 2 1 が複数ビットにより構成されていてももちろんよい。

【 0 2 9 4 】

このようなアプリケーションデータを生成するアプリケーションデータ生成部 3 4 の構成例を示すブロック図を図 3 2 に示す。

【 0 2 9 5 】

図 3 2 において、図 9 を参照して説明したモジュールについては、図 9 の場合と同じ番号を付してある。図 3 2 に示されるアプリケーションデータ生成部 4 3 0 の構成は、図 9 に示されるアプリケーションデータ生成部 3 4 の構成と基本的に同一であるが、図 3 2 に示されるアプリケーションデータ生成部 4 3 0 の場合、冗長データカウンタ 1 2 3 にオフセット値を供給するオフセット値供給部 1 2 2 が省略され、データ識別子を設定するデータ識別子設定部 4 3 1 が新たに設けられている。

20

【 0 2 9 6 】

冗長データ受付部 1 2 1 は、冗長データカウンタ 1 2 3 を初期化する際、カウント値を「 0 」に設定する。すなわち、オフセット値供給部 1 2 2 が存在しないので、初期化された冗長データカウンタ 1 2 3 のカウント値は「 0 」である。

【 0 2 9 7 】

データ識別子設定部 4 3 1 は、ヘッダ部 1 1 4 は、ヘッダ付加部に供給するための、データ識別子を設定し、設定したデータ識別子の設定値をヘッダ付加部 1 1 4 に供給する。ヘッダ付加部 1 1 4 は、アプリケーションデータの元データ / 冗長データ識別フィールド 4 2 1 に、供給されたデータ識別子の設定値を付加する。

30

【 0 2 9 8 】

図 3 2 のアプリケーションデータ生成部 4 3 0 により実行されるアプリケーションデータ生成処理を図 3 3 および図 3 4 のフローチャートを参照して説明する。

【 0 2 9 9 】

アプリケーションデータ生成部 4 3 0 の元データ受付部 1 1 1 および冗長データ受付部 1 2 1 は、ステップ S 1 8 1 において、図 1 8 のステップ S 6 1 の場合と同様に、各カウンタを初期化する。ただし、冗長データ受付部 1 2 1 は、冗長データカウンタ 1 2 3 のカウント値に、オフセット値を設定せず、他のカウンタと同様に、例えば「 0 」に設定する。

40

【 0 3 0 0 】

ステップ S 1 8 2 において、元データ受付部 1 1 1 は、図 1 8 のステップ S 6 2 の場合と同様に、元データを受け付けたか否かを判定し、受け付けたと判定した場合、図 1 8 のステップ S 6 3 乃至 S 6 5 の場合と同様に、ステップ S 1 8 3 において、受け付けた元データがブロック内の最初のパケットデータであるか否かを判定し、最初のパケットデータであると判定した場合、ステップ S 1 8 4 に処理を進め、ブロックカウンタ 1 1 3 をカウントアップさせ、ステップ S 1 8 5 において元データカウンタ 1 1 2 をカウントアップさ

50

せる。また、ステップS 1 8 3において、受け付けた元データがブロック内の最初のパケットデータでないと判定した場合、ステップS 1 8 4の処理を省略し、ステップS 1 8 5において、元データカウンタ1 1 2のみをカウントアップさせる。カウンタをカウントアップさせた元データ受付部1 1 1は、取得した元データをヘッダ付加部1 1 4に供給する。

【0301】

ステップS 1 8 6において、データ識別子設定部4 3 1は、ヘッダ付加部1 1 4が取得した元データに対して、データ識別子の値を「0」に設定し、ヘッダ付加部1 1 4に供給する。

【0302】

ヘッダ付加部1 1 4は、ステップS 1 8 7において、図1 8のステップS 6 6の場合と同様に、元データに元データカウンタ1 1 2のカウント値を付加する。また、ヘッダ付加部1 1 4は、ステップS 1 8 8において、データ識別子設定部4 3 1より供給されたデータ識別子を元データに付加する。さらに、ヘッダ付加部1 1 4は、ステップS 1 8 9において、図1 8のステップS 6 6の場合と同様に、元データにブロックカウンタ1 1 3のカウント値を付加する。そして、元データに必要なヘッダを付加したヘッダ付加部1 1 4は、ステップS 1 9 0において、図1 8のステップS 6 7の場合と同様に、それをアプリケーションデータとして、UDPパケット化部3 5に供給し、処理を図3 4のステップS 2 0 1に進める。

【0303】

なお、図3 3のステップS 1 8 2において、元データを受け付けていないと判定した場合、元データ受付部1 1 1は、ステップS 1 8 3乃至ステップS 1 9 0の処理を省略し、図3 4のステップS 2 0 1に処理を進める。

【0304】

図3 4のステップS 2 0 1において、冗長データ受付部1 2 1は、図1 8のステップS 6 8の場合と同様に、冗長データを受け付けたか否かを判定し、冗長データを受け付けたと判定した場合、図1 8のステップS 6 9の場合と同様に、ステップS 2 0 2において冗長データカウンタ1 2 3をカウントアップさせる。冗長データカウンタ1 2 3をカウントアップさせた冗長データ受付部1 1 1は、取得した冗長データをヘッダ付加部1 1 4に供給する。

【0305】

ステップS 2 0 3において、データ識別子設定部4 3 1は、ヘッダ付加部1 1 4が取得した冗長データに対して、データ識別子の値を「1」に設定し、ヘッダ付加部1 1 4に供給する。

【0306】

ヘッダ付加部1 1 4は、ステップS 2 0 4において、図1 8のステップS 7 0の場合と同様に、冗長データに冗長データカウンタ1 2 3のカウント値を付加する。また、ヘッダ付加部1 1 4は、ステップS 2 0 5において、データ識別子設定部4 3 1より供給されたデータ識別子を冗長データに付加する。さらに、ヘッダ付加部1 1 4は、ステップS 2 0 6において、図1 8のステップS 7 0の場合と同様に、冗長データにブロックカウンタ1 1 3のカウント値を付加する。そして、冗長データに必要なヘッダを付加したヘッダ付加部1 1 4は、ステップS 2 0 7において、図1 8のステップS 7 1の場合と同様に、それをアプリケーションデータとして、UDPパケット化部3 5に供給し、処理をステップS 2 0 8に進める。

【0307】

ステップS 2 0 8において、ヘッダ付加部1 1 4は、ブロックのデータを全て処理したか否かを判定し、未処理のパケットデータが存在すると判定した場合、図3 3ステップS 1 8 2に処理を戻し、それ以降の処理を繰り返す。また、図3 4のステップS 2 0 8において、全てのパケットデータを処理したと判定した場合、ヘッダ付加部1 1 4は、アプリケーションデータ生成処理を終了し、図1 5のステップS 5に処理を戻す。

10

20

30

40

50

【0308】

このように、識別子を用いて元データパケット番号と冗長データパケット番号を識別させることにより、受信装置13は、オフセット値を用いた場合と同様にパケットデータを分離することができる。

【0309】

以上においては、送信装置11と受信装置13は、互いに等しい所定の値を種データとして生成した同じ係数行列を用いて冗長データの作成処理およびロスした元データの復元処理を実行するように説明した。具体的には、送信装置11は、受信装置13がロスしたパケットを特定するための情報として、元データパケット番号を元データに付加し、受信装置13が取得したパケットを整列させるための情報として、冗長データパケット番号を冗長データパケット番号に付加して送信する。すなわち、送信装置11は、ロスしたパケットを特定するための情報を元データに付加し、冗長データを整列させるための情報を冗長データに付加して送信する。

10

【0310】

例えば、図35Aに示されるように、送信装置11は、所定の値を種データとして係数行列（行列番号1乃至3）を生成し、それを用いて冗長データ（冗長データパケット番号1乃至3）を算出し、それらを、それらのパケット番号順に、受信装置13に供給する。

【0311】

しかしながら、送信装置11がその係数行列を用いて生成し、送信した冗長データは、受信装置13において、その一部がロスしたり、受信順が、送信順と一致しなかったりする場合がある。例えば、図35Aにおいて、受信装置13は、冗長データパケット番号が「2」と「3」のパケットのみを受信する。このときの受信順も、送信順と異なり、冗長データは、冗長データパケット番号「3」、「2」の順番で受信される。

20

【0312】

受信装置13においても、係数行列は、送信装置11の場合と同様に所定の値を種データとして生成されるので、冗長データの受信順と係数行列の各行の並びは対応しない。従って、受信装置13は、それらに対応させるために、予め用意された係数行列に、受信した冗長データの順番を合わせるように、冗長データを整列させる。具体的には、受信装置13は、上述したように、冗長データパケット番号に基づいて、冗長データを整列させる。例えば、図35Aの場合、冗長データパケット番号が「3」、「2」の順番で受信した冗長データを、「2」、「3」の順に並べ替えて、冗長データの順番を係数行列に対応させる。このようにすることにより、受信装置13は、ロスした元データ（図35Aの場合、元データパケット番号「5」の元データ）を復元することができる。

30

【0313】

しかしながら、それ以外の方法を用いてもよく、例えば、送信装置11と受信装置13とで種データを予め共有せずに、送信装置11が、通信時にパケットとともに、その種データの情報（すなわち、係数行列を生成するための情報）を受信装置13に送信するようにしてもよい（図35B）。

【0314】

例えば、図35Bにおいては、送信装置11および受信装置13は、それぞれ、冗長データパケット番号を用いて種データを生成している。このとき、送信装置11および受信装置13は、1つの冗長データパケット番号を種データとして1行の係数行列を生成し、各行を合成して係数行列を完成させる。このようにすることにより、受信装置13は、受信した冗長データの受信順で冗長データパケット番号を種データとするので、必要な分だけ係数行列（受信した冗長データに対応する行のみの係数行列）を生成することができる。また、その生成された係数行列の行の順番は、受信した冗長データの受信順に対応することになる。

40

【0315】

すなわち、送信装置11は、ロスしたパケットを特定するための情報（元データパケット番号）を元データに付加し、冗長データに対応した係数行列を生成させるための情報（

50

冗長データ packets 番号) を冗長データに付加して送信する。

【0316】

なお、その場合の送信装置 11 の処理は、種データとして、所定の値の代わりに、これから生成する予定の冗長データ packets 番号を用いるだけで、基本的には、上述した処理と同様の処理を実行するので、その説明を省略する。受信装置 13 も、上述した場合と基本的に同様にデータ受信処理を行う。ただし、ロス packets 復元部 228 は、元データ復元処理を実行する際に、図 20 の場合と異なる処理を実行する。

【0317】

図 36 のフローチャートを参照して、ロス packets 復元部 228 による元データ復元処理を説明する。

10

【0318】

図 14 のロス packets 復元部 228 の行列式作成部 253 は、判定結果取得部 251 からの指示に基づいて、元データ復元処理を開始し、ステップ S221 において、元データ保持部 252 より元データのアプリケーションデータを取得し、ステップ S222 において、取得したアプリケーションデータの元データ packets 番号を参照して、ロスした元データの packets 番号を特定する。また、行列式作成部 253 は、ステップ S223 において、冗長データ保持部 254 より冗長データのアプリケーションデータを取得する。これらの処理は、図 20 のステップ S121 乃至ステップ S123 にそれぞれ対応する。

【0319】

行列式作成部 253 は、取得したアプリケーションデータの冗長データ packets 番号を参照し、その冗長データ packets 番号を係数行列供給部 255 に供給する。係数行列供給部 255 は、ステップ S224 において、供給された冗長データ packets 番号を種データとして係数行列を生成し、それを行列式作成部 253 に供給する。この係数行列は、受信装置 13 が取得した冗長データに対応する行と、全ての元データに対応する列とで構成される M - L 行 N 列の係数行列である。

20

【0320】

次に、行列式作成部 253 は、ステップ S225 において、取得した元データの packets 番号に基づいて、ステップ S224 において生成された係数行列より必要な成分を取得する。すなわち、行列式作成部 253 は、M - L 行 N 列の係数行列より、取得した元データに対応する列成分を抽出し、M - L 行 K 列の係数行列とする。

30

【0321】

ステップ S226 において、行列式作成部 253 は、計算用冗長データを求める行列式を作成し、それを行列式演算部 256 に供給する。そして、ステップ S227 において、供給された行列式を演算し、計算用冗長データを算出し、算出した計算用冗長データを行列式作成部 253 に戻す。なおこれらの処理は図 20 のステップ S126 の処理、およびステップ S127 の処理を終了する。

【0322】

計算用冗長データを取得した行列式作成部 253 は、ステップ S228 において、ロスした元データの packets 番号と、取得した冗長データの packets 番号に基づいて、ステップ S224 において生成された係数行列 (M - L 行 N 列) より必要な部分 (M - L 行 K 列) を取得し、ステップ S229 において、ロスした元データを求めるための行列式 (例えば、式 (4)) を作成し、作成した行列式を行列式演算部 256 に供給する。なおこれらの処理は図 20 のステップ S128 の処理、およびステップ S129 の処理を終了する。

40

【0323】

行列式演算部 256 は、ステップ S230 において、求解処理を実行し、EOR 演算によって供給された行列式の解を求め、ステップ S231 においてその行列式の解を元データとして出力し、元データ復元処理を終了して、図 19 のステップ S99 に処理を戻す。

【0324】

行列式作成部 253 が、以上のように元データ復元処理を行うことにより、受信装置 13 は、送信装置 11 より供給される冗長データに付加された冗長データ packets 番号を種

50

データとして係数行列を生成し、その係数行列を用いてロスした元データを復元することができる。これにより、受信装置 13 は、より容易、かつ、より正確にデータを受信することができる。

【0325】

以上において、送信装置 11 は、受信装置 13 がロスした元データを特定する情報として元データパケット番号を元データに付加して送信するように説明したが、受信装置 13 がロスした元データを特定する情報として他の情報を元データに付加するようにしてももちろんよい。また、送信装置 11 は、この受信装置 13 がロスした元データを特定する情報を、元データに付加せず、アプリケーションデータとは異なるデータとして送信するようにしてもよい。さらに、送信装置 11 は、この受信装置 13 がロスした元データを特定する情報を、例えば、磁気ディスクや書き込み可能な光ディスク等のような記録媒体等を用いて受信装置 13 に供給されるようにしてもよい。

10

【0326】

また、以上において、送信装置 11 は、受信装置 13 が取得した冗長データを整列させる情報として冗長データパケット番号を冗長データに付加して送信するように説明したが、受信装置 13 が取得した冗長データを整列させる情報として他の情報を冗長データに付加するようにしてももちろんよい。また、送信装置 11 は、この受信装置 13 が取得した冗長データを整列させる情報を冗長データに付加せず、アプリケーションデータとは異なるデータとして送信するようにしてもよい。さらに、送信装置 11 は、この受信装置 13 が取得した冗長データを整列させる情報を、例えば、磁気ディスクや書き込み可能な光ディスク等のような記録媒体等を用いて受信装置 13 に供給されるようにしてもよい。

20

【0327】

さらに、以上において、送信装置 11 は、受信装置 13 が係数行列を生成する情報として冗長データパケット番号を冗長データに付加して送信するように説明したが、受信装置 13 が係数行列を生成する情報として他の情報を冗長データに付加するようにしてももちろんよい。また、送信装置 11 は、この受信装置 13 が係数行列を生成する情報を冗長データに付加せず、アプリケーションデータとは異なるデータとして送信するようにしてもよい。さらに、送信装置 11 は、この受信装置 13 が係数行列を生成する情報を、例えば、磁気ディスクや書き込み可能な光ディスク等のような記録媒体等を用いて受信装置 13 に供給されるようにしてもよい。

30

【0328】

なお、送信装置 11 は、受信装置 13 が取得したパケットデータがどのブロックに属するデータであるかを特定するための情報、受信装置 13 がロスした元データを特定するための情報、受信装置 13 が取得した冗長データを整列させるための情報、および、受信装置 13 が係数行列を生成するための情報等の復元処理に関する情報である復元関連情報の内、2つ以上を1つのパケットデータに付加して送信するようにしてもよい。また、同じ元データであっても、これらの情報を付加するパケットデータと、付加しないパケットデータを混在させるようにしてもよい。冗長データについても同様である。

【0329】

また、復元性能は上述した方法より低いものの、係数行列をブロック毎に変更するようにしてもよい。その場合、予め複数の係数行列を、送信装置 11 および受信装置 13 の両方に用意するようにしてもよいし、種データをブロック毎に変更するようにしてもよい。種データを用いて係数行列を生成する場合、その種データとして、冗長データパケット番号を用いるように説明したが、種データをブロック毎に変更する場合、例えば、冗長データパケット番号の他にブロック番号も種データとして利用すれば、生成する行列を容易に変更することができる。

40

【0330】

また、ブロックデータのデータサイズは、ブロック間で共通であっても良いし、互いに異なるようにしてもよい。ブロックデータのデータサイズが互いに異なるようにする場合、各ブロックのデータサイズ(または、各ブロックに対応するパケットデータ数)が受信

50

装置 1 3 において把握することができるように、何らかの情報を送信装置 1 1 より受信装置 1 3 に供給する。

【 0 3 3 1 】

以上においては、通信システム 1 が、データを送信する送信装置 1 1 およびデータを受信する受信装置 1 3 により構成されるように説明したが、通信システムの構成は、これに限らず、通信システムは、例えば、図 3 7 に示されるように、図 1 の送信装置 1 1 および受信装置 1 3 を、それぞれモジュールとし、送信部および受信部とし、それらの両方を含む 2 台の送受信装置により構成されるようにしてもよい。

【 0 3 3 2 】

図 3 7 において、通信システム 5 0 1 は、図 1 のネットワーク 1 2 と同様の、ネットワーク 5 1 2 に接続された送受信装置 5 1 1 および送受信装置 5 1 3 により構成される。送受信装置 5 1 1 は、送信部 5 2 1 および受信部 5 2 2 を含み、送受信装置 5 1 3 は、送信部 5 3 1 および受信部 5 3 2 を含む。送信部 5 2 1 および送信部 5 3 1 は、図 1 の送信装置 1 1 と基本的に同様の構成であり、同様に動作するのでその詳細な説明は省略する。

10

【 0 3 3 3 】

また、図 3 7 において、受信部 5 2 2 および受信部 5 3 2 は、図 1 の受信装置 1 3 と基本的に同様の構成であり、同様に動作するのでその詳細な説明は省略する。

【 0 3 3 4 】

すなわち、図 3 7 の通信システム 5 0 1 は、内蔵する送信部および受信部をそれぞれ有する送受信装置 5 1 1 および送受信装置 5 1 3 により構成され、それらが互いに上述したようなEOR演算を用いた冗長データを利用して、再送制御なしに、通信処理を行う。

20

【 0 3 3 5 】

このようにすることにより、双方向の通信においても、受信部は、高速であり、かつ、復元性能の高い復元処理を、製造コストを増大せずを実現することができる。従って、送受信装置 5 1 1 および送受信装置 5 1 3 は、より容易、かつ、より正確にデータを受信することができる。

【 0 3 3 6 】

なお、図 1 においては、通信システム 1 は、1 台の送信装置 1 1 および 1 台の受信装置 1 3 により構成されるように説明したが、これに限らず、複数台の送信装置または複数台の受信装置を含むようにしてもよい。

30

【 0 3 3 7 】

図 3 8 は、本発明を適用したコンテンツ配信システムの構成例を示す図である。コンテンツ配信システム 6 0 1 のサーバ 6 1 1 は、図 1 のネットワークと同等のネットワーク 6 1 2 に接続され、同様にネットワーク 6 1 2 に接続された複数の端末装置 6 1 3 および 6 1 4 により構成される。

【 0 3 3 8 】

図 3 8 において、サーバ 6 1 1 は、送信装置 1 1 と同様の機能を有しており、ネットワーク 6 1 2 を介して、端末装置 6 1 3 および端末装置 6 1 4 のそれぞれに対して、画像データや音声データ等のコンテンツデータをパケット化し、上述したようにEOR演算による冗長パケットを付加し、そのパケットをマルチキャストに送信する。

40

【 0 3 3 9 】

端末装置 6 1 3 および端末装置 6 1 4 は、それぞれ、受信装置 1 3 と同様の機能を有しており、サーバ 6 1 1 より供給されるパケットをそれぞれ受信し、上述したようにロスしたパケットを、冗長データを用いて復元する等して、コンテンツデータを復元し、再生する。

【 0 3 4 0 】

なお、サーバ 6 1 1 は、コンテンツデータをストリーミング配信するようにしてもよいし、ダウンロードしてから再生させるように配信するようにしてもよい。

【 0 3 4 1 】

以上のように、サーバ 6 1 1 が複数台の端末装置 6 1 3 および 6 1 4 に対してパケット

50

を配信する場合、端末装置 6 1 3 および端末装置 6 1 4 がパケットをロスしたときに、そのパケットの再送要求をサーバ 6 1 1 にするようにすると、サーバ 6 1 1 に再生要求が集中し、サーバ 6 1 1 の負荷が大きくなり、ダウンしてしまう恐れがある。従って、サーバ 6 1 1 が、上述したように、負荷の軽いEOR演算による冗長データを生成し、それをコンテンツデータとともに配信することにより、端末装置 6 1 3 および端末装置 6 1 4 は、パケットをロスしても、再生要求をせずに、コンテンツデータを復元することができる。従って、サーバ 6 1 1 における負荷の増大を抑制することができ、安定したコンテンツデータの配信サービスを提供することができる。

【 0 3 4 2 】

なお、上述したEOR演算による冗長データを送信データに付加する方法と、再送要求を用いる方法とを組み合わせる利用するようにしてももちろんよい。その場合、受信側の装置は、パケットロスが発生した場合、まず、受信した冗長データにより元データの復元を試み、復元不能であった場合に、送信元に対してロスしたパケットの再送要求を行うようにする。このようにすることにより、再送要求の回数を削減することができるので、送信側と受信側の両方の装置において、再送要求による処理の負荷を軽減させることができる。すなわち、システム全体の負荷を削減することができる。

10

【 0 3 4 3 】

また、図 1 の通信システム 1 において、送信装置 1 1 と受信装置 1 3 が接続されるネットワーク 1 2 は、その一部または全部が有線により構成されるようにしてもよいし、無線により構成されるようにしてもよい。またネットワーク 1 2 が複数のネットワークにより構成されるようにしてもよい。すなわち、送信装置 1 1 と受信装置 1 3 とが複数のルータ等を介して接続されるようにしても良い。

20

【 0 3 4 4 】

さらに、送信装置 1 1 および受信装置 1 3 は、上述した構成以外の構成を含むようにしてもよい。例えば、モニタ等の表示部を設けるようにしてもよいし、キーボードやボタン等の入力部を設けるようにしてもよい。

【 0 3 4 5 】

上述した一連の処理は、ハードウェアにより実行させることもできるし、ソフトウェアにより実行させることもできる。この場合、例えば、図 1 の送信装置 1 1 や受信装置 1 3 には、図 3 9 に示されるようなパーソナルコンピュータが含まれる。

30

【 0 3 4 6 】

図 3 9 において、パーソナルコンピュータ 7 0 1 のCPU 7 1 1 は、ROM (Read Only Memory) 7 1 2 に記憶されているプログラム、または記憶部 7 2 3 からRAM 7 1 3 にロードされたプログラムに従って各種の処理を実行する。RAM 7 1 3 にはまた、CPU 7 1 1 が各種の処理を実行する上において必要なデータなども適宜記憶される。

【 0 3 4 7 】

CPU 7 1 1、ROM 7 1 2、およびRAM 7 1 3 は、バス 7 1 4 を介して相互に接続されている。このバス 7 1 4 にはまた、入出力インタフェース 7 2 0 も接続されている。

【 0 3 4 8 】

入出力インタフェース 7 2 0 には、キーボード、マウスなどよりなる入力部 7 2 1、CRT、LCDなどよりなるディスプレイ、並びにスピーカなどよりなる出力部 7 2 2、ハードディスクなどより構成される記憶部 7 2 3、モデムなどより構成される通信部 7 2 4 が接続されている。通信部 7 2 4 は、インターネットを含むネットワークを介しての通信処理を行う。

40

【 0 3 4 9 】

入出力インタフェース 7 2 0 にはまた、必要に応じてドライブ 7 2 5 が接続され、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリなどのリムーバブルメディア 7 2 6 が適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部 7 2 3 にインストールされる。

【 0 3 5 0 】

50

上述した一連の処理をソフトウェアにより実行させる場合には、そのソフトウェアを構成するプログラムが、ネットワークや記録媒体からインストールされる。

【0351】

この記録媒体は、例えば、図39に示されるように、装置本体とは別に、ユーザにプログラムを配信するために配布される、プログラムが記録されている磁気ディスク（フレキシブルディスクを含む）、光ディスク（CD-ROM(Compact Disk-Read Only Memory),DVD(Digital Versatile Disk)を含む）、光磁気ディスク（MD(Mini-Disk)（登録商標）を含む）、もしくは半導体メモリなどよりなるリムーバブルメディア726により構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに配信される、プログラムが記録されているROM712や、記憶部723に含まれるハードディスクなどで構成される。

10

【0352】

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0353】

また、本明細書において、システムとは、複数の装置により構成される装置全体を表すものである。

【図面の簡単な説明】

【0354】

【図1】本発明を適用した通信システムの構成例を示す図である。

20

【図2】図1の送信装置の構成例を示すブロック図である。

【図3】図1の通信システムにおけるデータの処理の流れを示す図である。

【図4】UDPパケットの構成例を示す模式図である。

【図5】図2の冗長データ生成部の詳細な構成例を示すブロック図である。

【図6】パケットデータの構成例を示す模式図である。

【図7】図5の係数行列供給部の詳細な構成例を示すブロック図である。

【図8】係数行列の生成方法と復元能力の関係の例を示す図である。

【図9】図2のアプリケーションデータ生成部の詳細な構成例を示すブロック図である。

【図10】アプリケーションデータの構成例を示す模式図である。

【図11】パケット番号の例を示す図である。

30

【図12】図1の受信装置の構成例を示すブロック図である。

【図13】パケットをロスする様子 of 例を示す図である。

【図14】図12のロスパケット復元部の詳細な構成例を示すブロック図である。

【図15】データ送信処理の例を説明するフローチャートである。

【図16】冗長データ生成処理の例を説明するフローチャートである。

【図17】係数行列準備処理の例を説明するフローチャートである。

【図18】アプリケーションデータ生成処理の例を説明するフローチャートである。

【図19】データ受信処理の例を説明するフローチャートである。

【図20】元データ復元処理の例を説明するフローチャートである。

【図21】求解処理の例を説明するフローチャートである。

40

【図22】求解処理の例を説明する、図21に続くフローチャートである。

【図23】ガウス消去法の手順の例を説明する図である。

【図24】ガウス消去法の手順の例を説明する図である。

【図25】ガウス消去法の手順の例を説明する図である。

【図26】ガウス消去法の手順の例を説明する図である。

【図27】係数生成確率と復元能力の関係を説明する図である。

【図28】係数生成確率と復元能力の関係を説明する図である。

【図29】EOR演算による復元処理と、リードソロモン符号を用いた復元処理との、処理速度の比較結果の例を示す図である。

【図30】EOR演算による復元処理と、リードソロモン符号を用いた復元処理との、エラ

50

一率の比較結果の例を示す図である。

【図31】アプリケーションデータの、他の構成例を示す模式図である。

【図32】図2のアプリケーションデータ生成部の、他の構成例を示すブロック図である。

【図33】アプリケーションデータ生成処理の、他の例を説明するフローチャートである。

【図34】アプリケーションデータ生成処理の、他の例を説明する、図32に続くフローチャートである。

【図35】係数行列の生成方法の、他の例を説明する模式図である。

【図36】元データ復元処理の、他の例を説明するフローチャートである。

【図37】本発明を適用した通信システムの、他の構成例を示す図である。

【図38】本発明を適用したコンテンツ配信システムの構成例を示す図である。

【図39】パーソナルコンピュータの構成例を示すブロック図である。

【符号の説明】

【0355】

1 通信システム, 11 送信装置 12 ネットワーク, 13 受信装置, 31 データ分割部, 32 ブロック分割部, 33 冗長データ生成部, 34 アプリケーションデータ生成部, 35 UDPパケット化部, 36 バッファ, 37 出力制御部, 38 IP処理部, 39 イーサネット(R)処理部, 40 ケーブル, 51 送信データ, 52 ブロック, 53 元データ, 54 冗長データ, 55 送信パケット, 71 元データ保持部, 72 行列式生成部, 73 係数行列供給部, 74 行列式演算部, 75 解出力部, 81 元データ, 82 冗長データ, 91 種データ保持部, 92 擬似乱数生成部, 93 擬似乱数バッファ, 94 係数判定部, 95 係数生成確率保持部, 96 係数行列保持部, 97 係数行列出力部, 111 元データ受付部, 112 元データカウンタ, 113 ブロックカウンタ, 114 ヘッダ付加部, 121 冗長データ受付部, 122 オフセット値供給部, 123 冗長データカウンタ, 131 アプリケーションデータ, 132 ブロック番号, 133 元データパケット番号, 134 元データ, 141 アプリケーションデータ, 142 ブロック番号, 143 冗長データパケット番号, 144 冗長データ, 210 ケーブル, 221 イーサネット(R)処理部, 222 IP処理部, 223 アプリケーションデータ抽出部, 224 バッファ, 225 データ分離部, 226 パケットロス判定部, 227 ブロックデータ復元部, 228 ロスパケット復元部, 229 ブロック合成部, 241 元データ, 242 冗長データ, 251 判定結果取得部, 252 元データ保持部, 253 行列式作成部, 254 冗長データ保持部, 255 係数行列供給部, 256 行列式演算部, 257 解出力部, 430 アプリケーションデータ生成部, 431 データ識別子設定部, 501 通信システム, 511 送受信装置, 512 ネットワーク, 513 送受信装置, 521 送信部, 522 受信部, 531 送信部, 532 受信部, 601 コンテンツ配信システム, 611 サーバ, 612 ネットワーク, 613 端末装置, 614 端末装置, 701 パーソナルコンピュータ, 711 CPU, 712 ROM, 713 RAM, 714 バス, 720 入出力インタフェース, 721 入力部, 722 出力部, 723 記憶部, 724 通信部, 725 ドライブ, 726 リムーバブルメディア

10

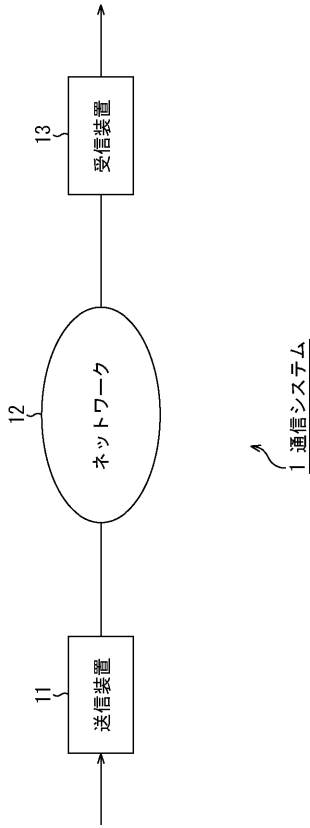
20

30

40

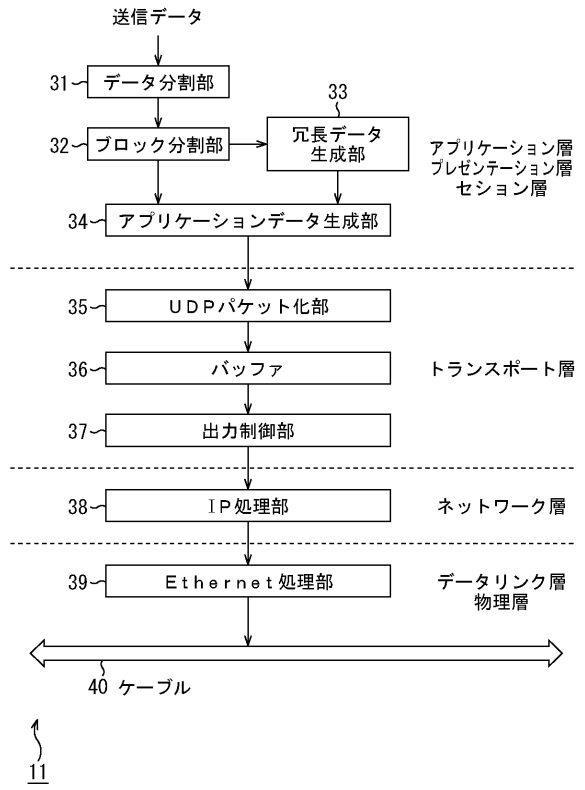
【図1】

図1



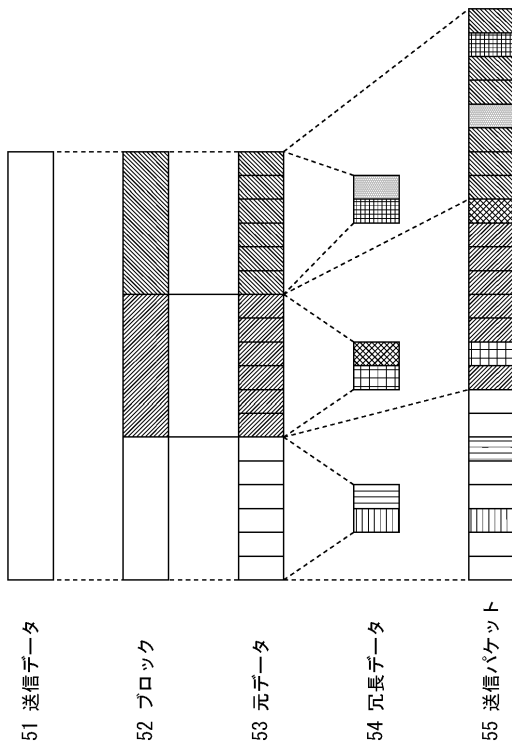
【図2】

図2



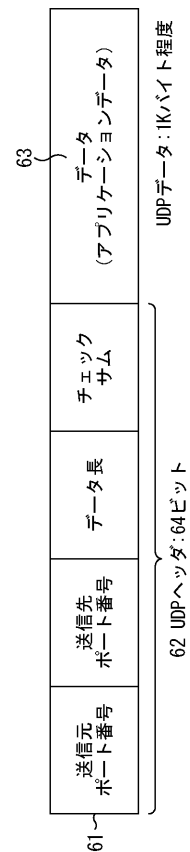
【図3】

図3



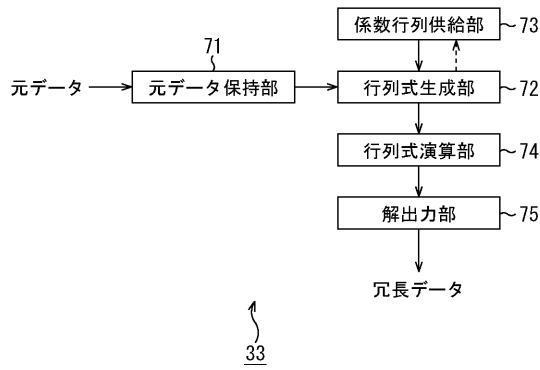
【図4】

図4



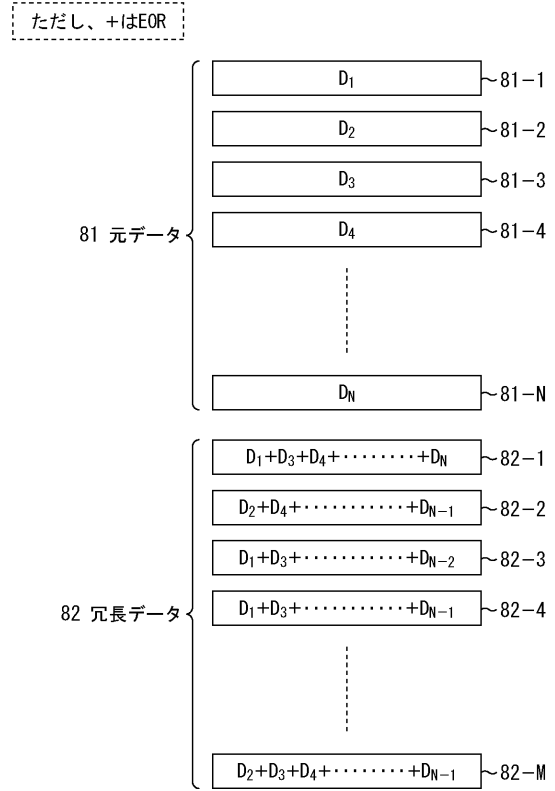
【図5】

図5



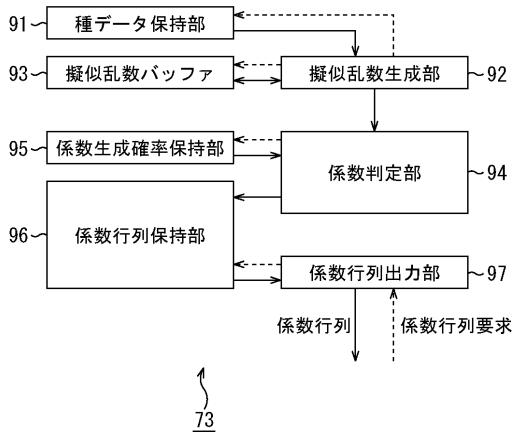
【図6】

図6



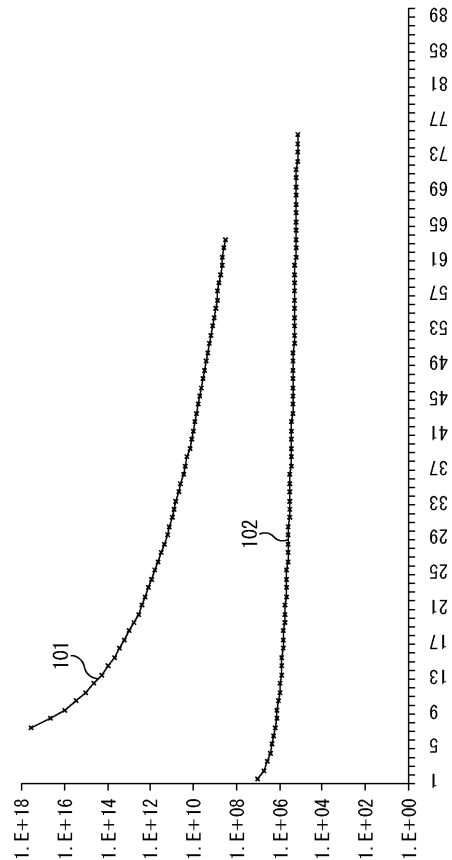
【図7】

図7



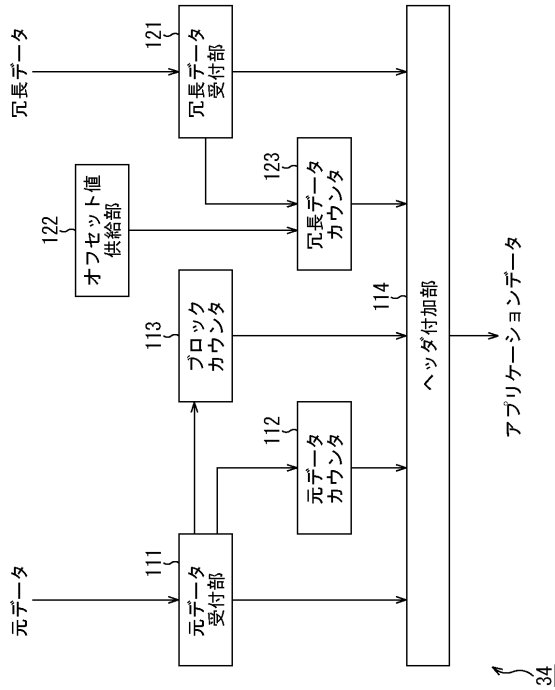
【図8】

図8



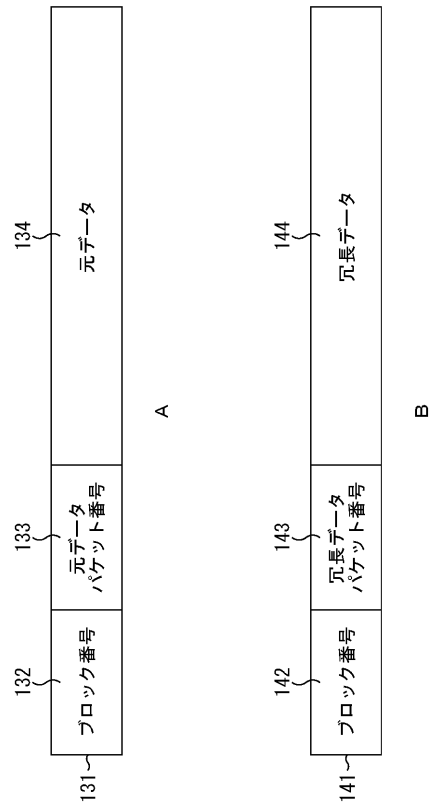
【図9】

図9



【図10】

図10



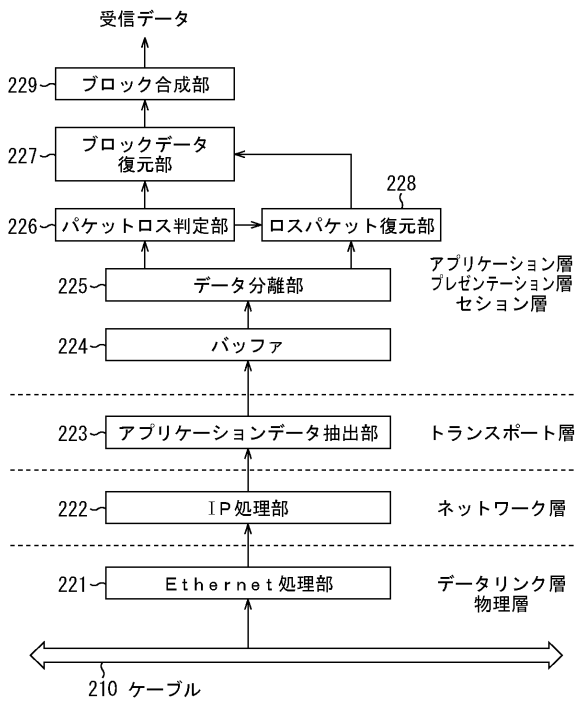
【図11】

図11

ブロック番号	パケット番号	元データ/冗長データ
0	0	元データ
0	1	元データ
0	2	元データ
0	3	元データ
-	-	元データ
0	999	元データ
0	1000	冗長データ
0	1001	冗長データ
-	-	冗長データ
0	1099	冗長データ

【図12】

図12

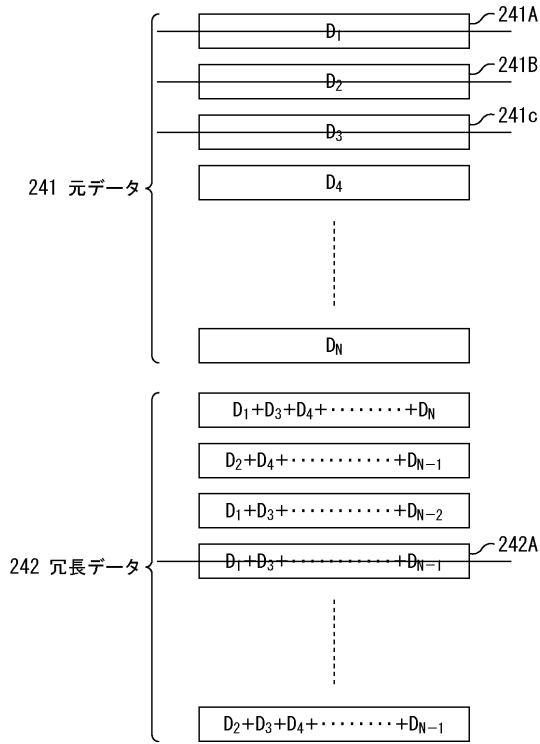


13

【図13】

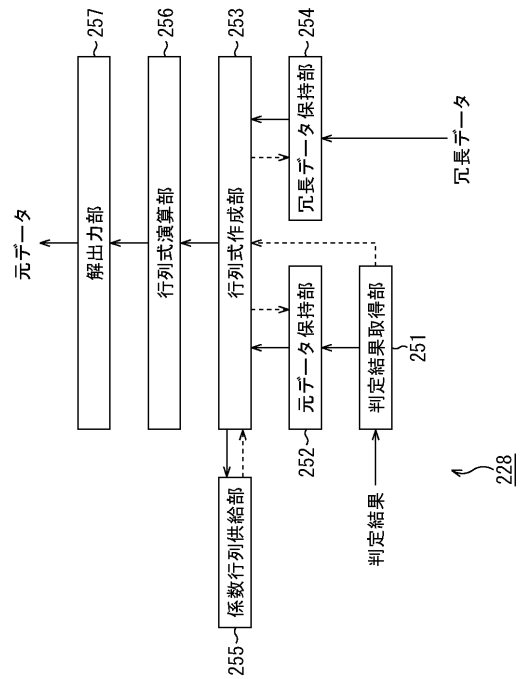
図13

ただし、+はEOR



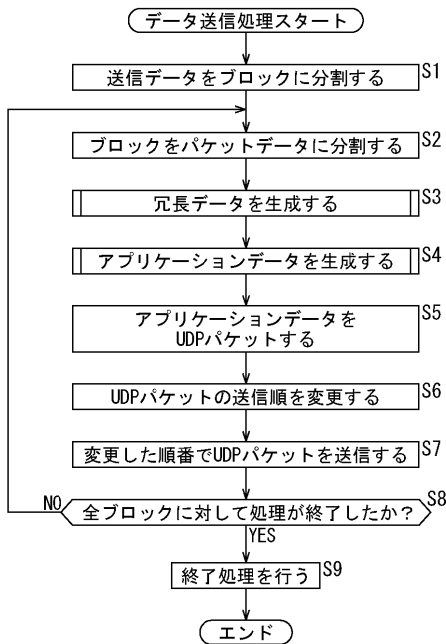
【図14】

図14



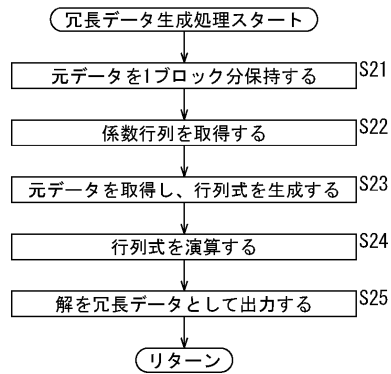
【図15】

図15



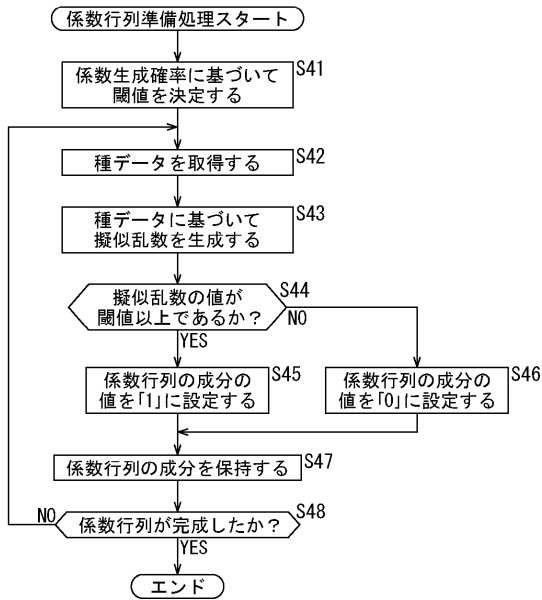
【図16】

図16



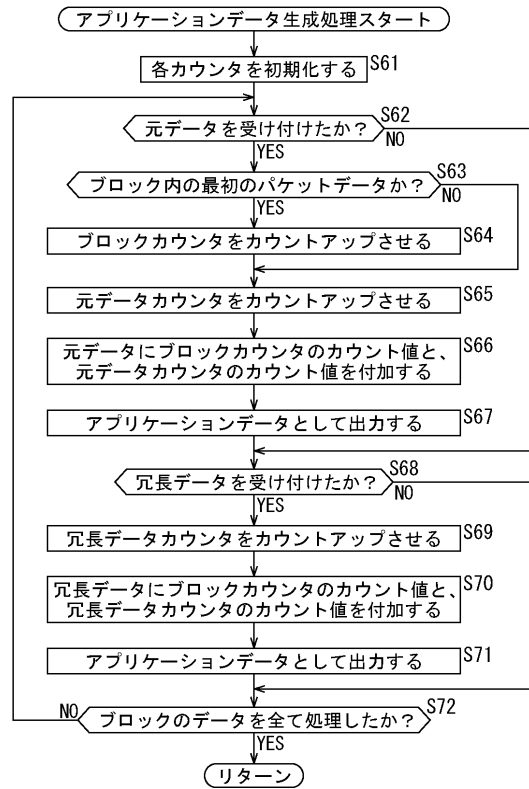
【図 17】

図17



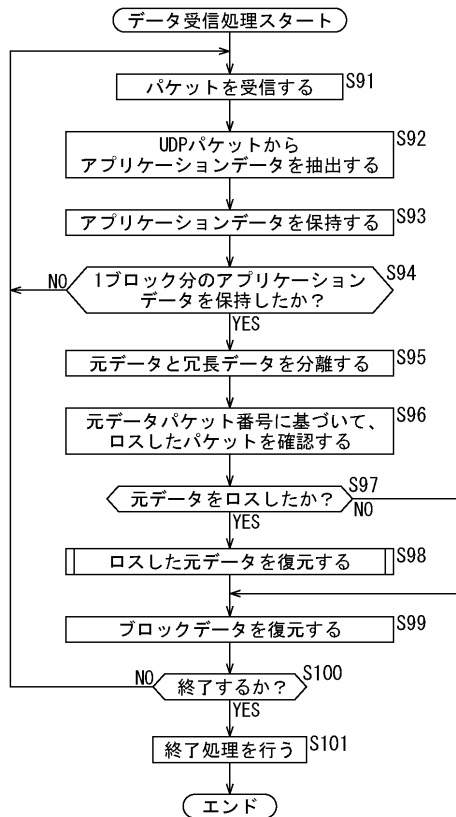
【図 18】

図18



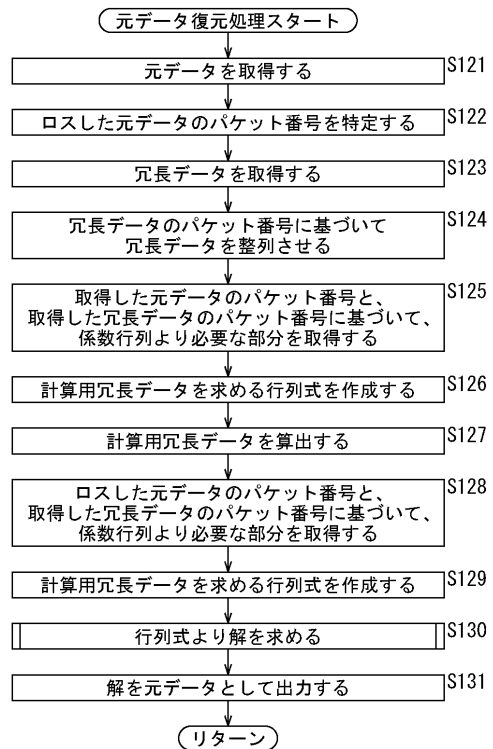
【図 19】

図19

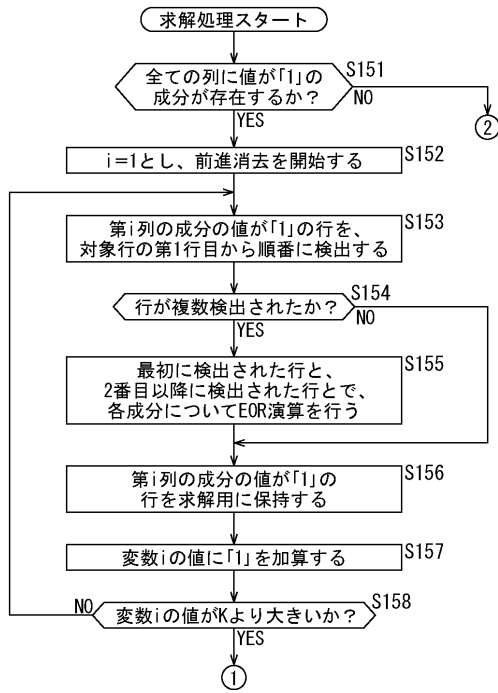


【図 20】

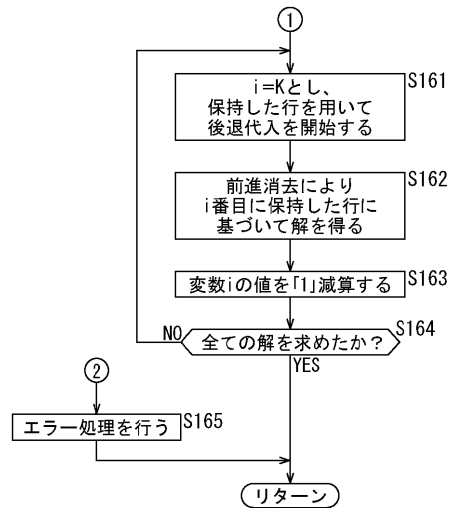
図20



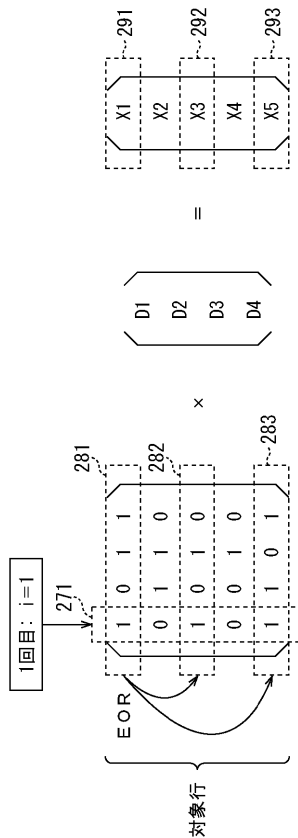
【図 2 1】
図21



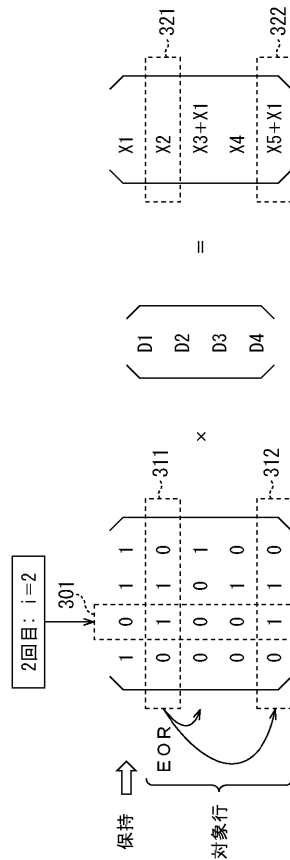
【図 2 2】
図22



【図 2 3】
図23

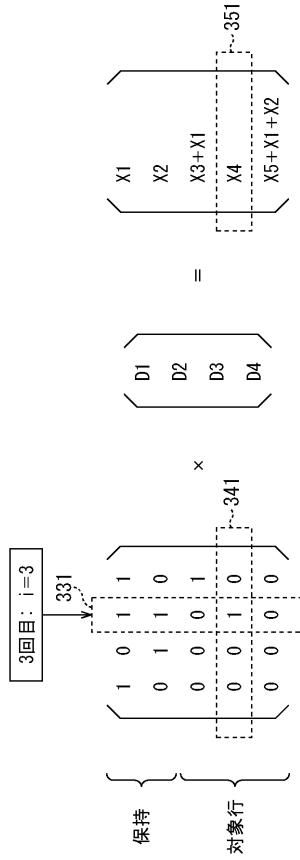


【図 2 4】
図24



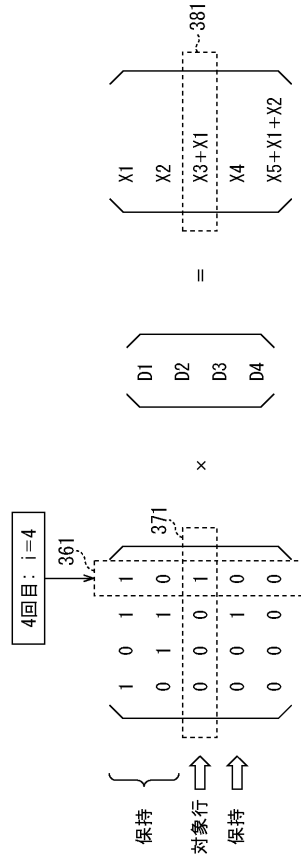
【図 25】

図25



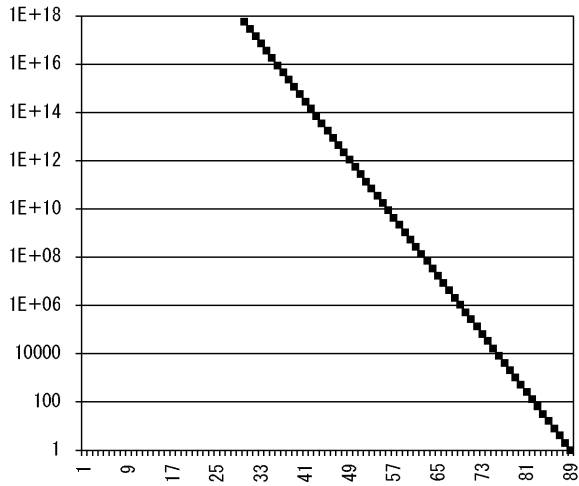
【図 26】

図26



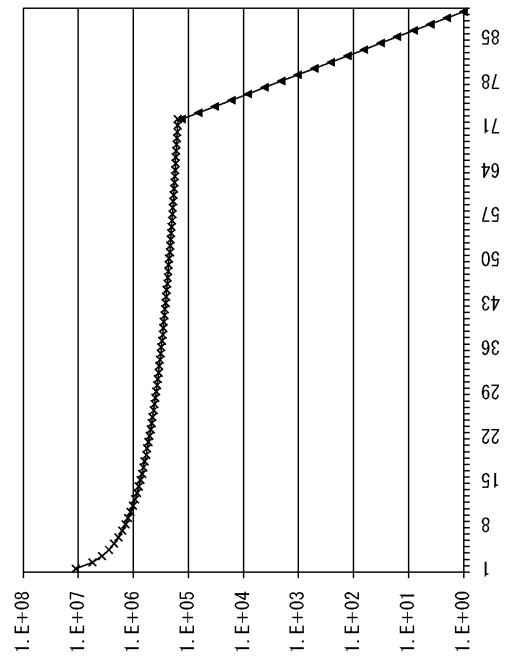
【図 27】

図27



【図 28】

図28



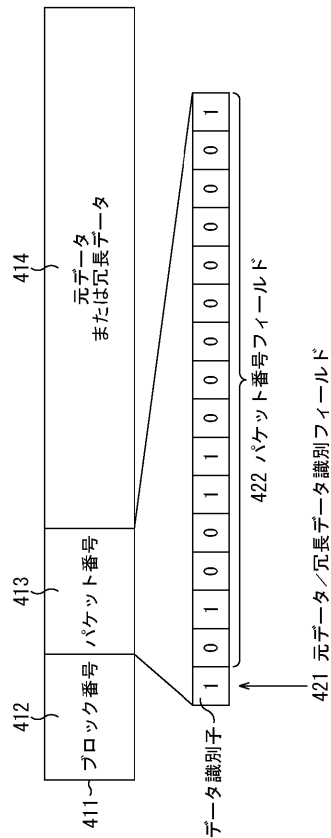
【図 29】
図29

391		リードソロモンFEC (参考値)
前処理	EOR	8600 ms
ガウスの消去		600 ms
データ復元		400 ms
総計		240 ms (4.8%)
		9600 ms (190%)

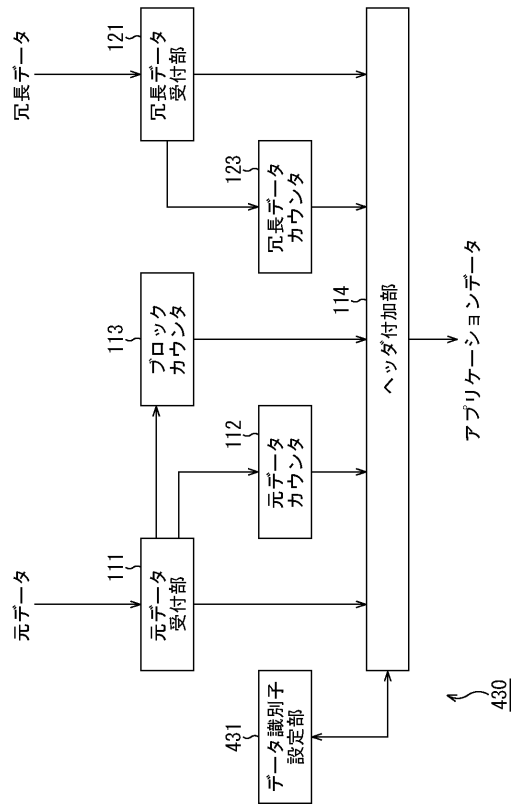
【図 30】
図30

401		EOR		RS-FEC
パケット数/ブロック	1000	2000	100	
冗長パケット数	100	200	10	
エラー率が 10^{-3} となるパケットロス率	5.0%	6.1%	1.0%	
エラー率が 10^{-14} となるパケットロス率	3.5%	5.0%	0.3%	

【図 31】
図31

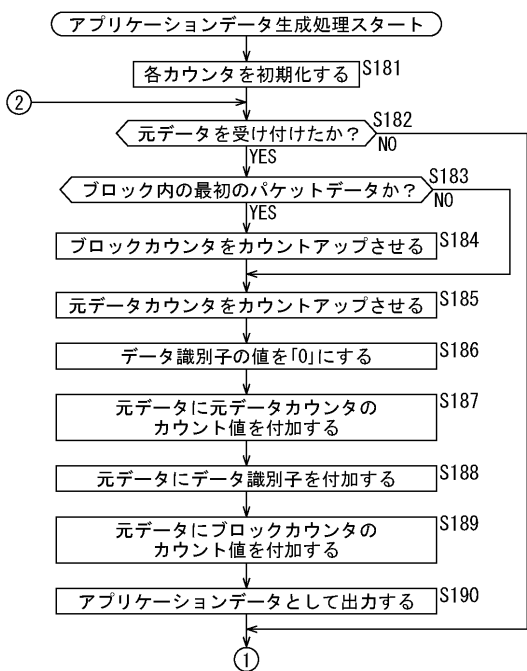


【図 32】
図32



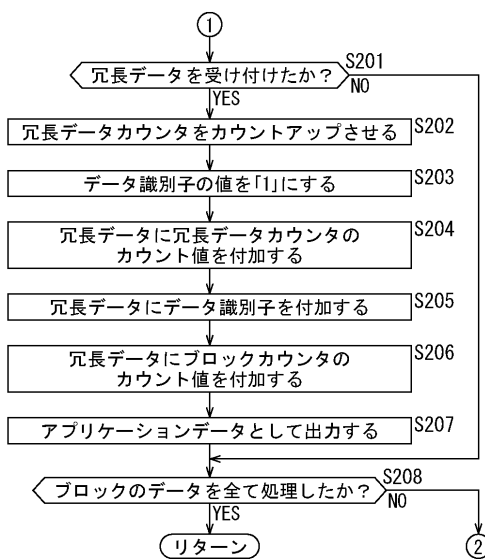
【図 3 3】

図33



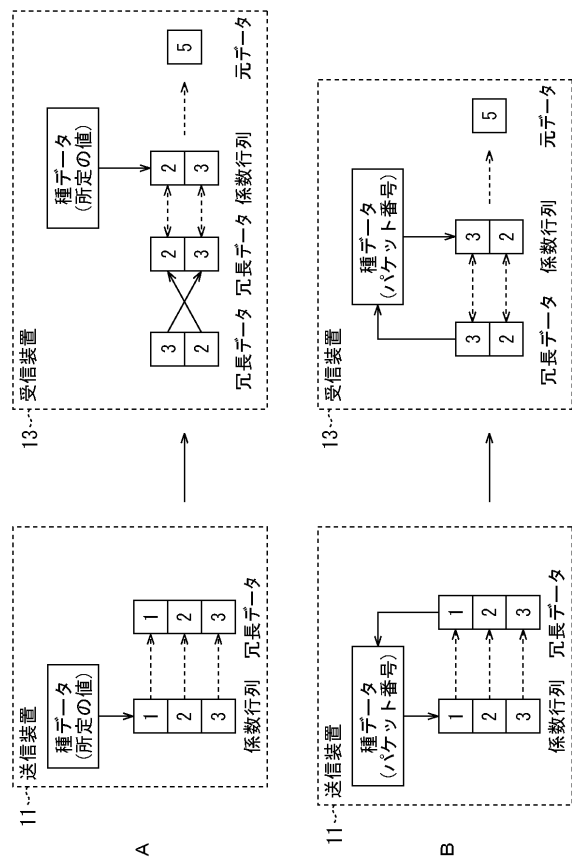
【図 3 4】

図34



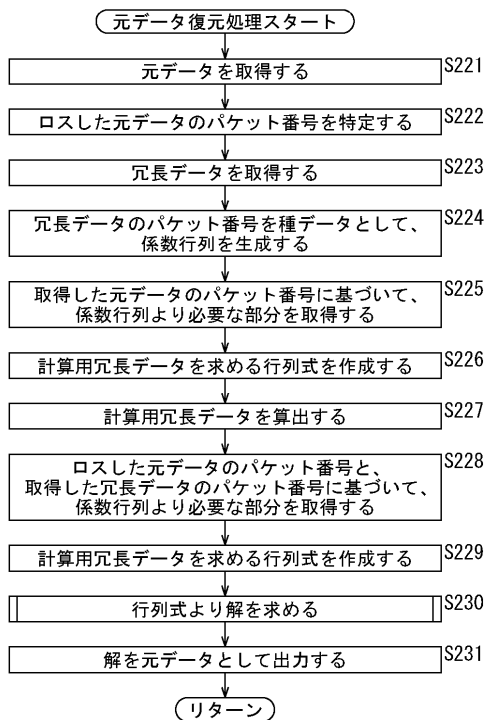
【図 3 5】

図35

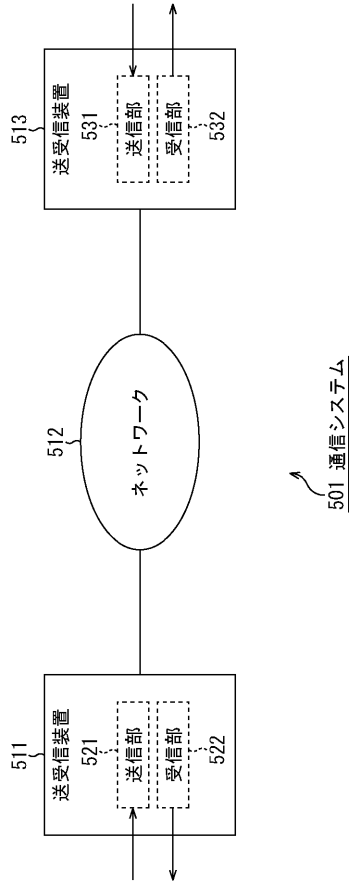


【図 3 6】

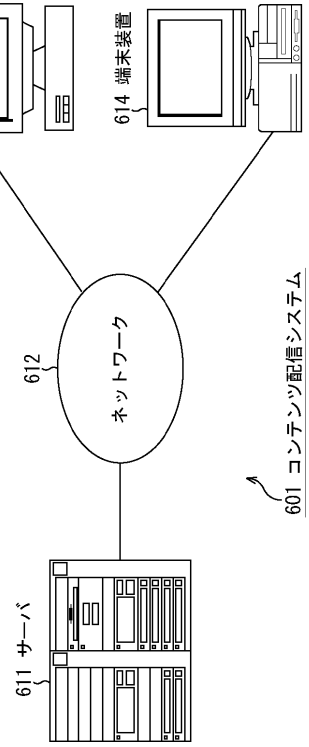
図36



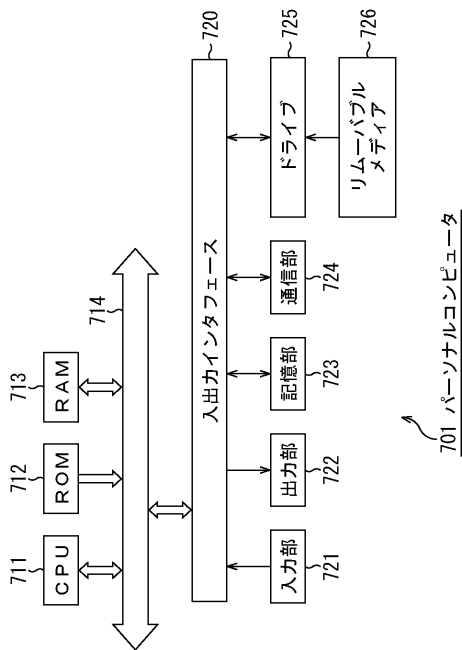
【図37】
図37



【図38】
図38



【図39】
図39



フロントページの続き

(56)参考文献 特開2003-258776(JP,A)

特開平11-136220(JP,A)

特開平09-008673(JP,A)

特開平07-143100(JP,A)

John W. Byers, et al., A Digital Fountain Approach to Reliable Distribution of Bulk Data, Proceedings of ACM SIGCOMM '98, 1998年 9月, p.56-67

山本 篤史, 他, 畳込み符号を用いたバーストパケット損の回復, 電子情報通信学会技術研究報告, 日本, 社団法人電子情報通信学会, 2001年 7月18日, Vol.101 No.215, p.9-16

千田 陽介, 他, DirectShareによるコンピュータ作業環境の携帯とデータ共有, 情報処理学会研究報, 日本, 社団法人情報処理学会, 2003年11月19日, Vol.2003 No.115, p.159-164

(58)調査した分野(Int.Cl., DB名)

H04L 1/00

G06F 11/10

H03M 13/19