

(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号  
特許第7408152号  
(P7408152)

(45)発行日 令和6年1月5日(2024.1.5)

(24)登録日 令和5年12月22日(2023.12.22)

(51)国際特許分類

F I

H 0 3 M 13/35 (2006.01)

H 0 3 M 13/35

H 0 4 L 1/00 (2006.01)

H 0 4 L 1/00

B

E

請求項の数 13 (全42頁)

(21)出願番号	特願2020-500808(P2020-500808)	(73)特許権者	507190994
(86)(22)出願日	平成30年7月9日(2018.7.9)		上海交通大学
(65)公表番号	特表2020-526979(P2020-526979 A)		SHANGHAI JIAO TONG UNIVERSITY
(43)公表日	令和2年8月31日(2020.8.31)		中華人民共和国上海市閔行区東川路800号
(86)国際出願番号	PCT/CN2018/095051		800 Dongchuan Rd., Minhang District, Shanghai, 200240, P.R. CHINA
(87)国際公開番号	WO2019/011219	(74)代理人	100205936
(87)国際公開日	平成31年1月17日(2019.1.17)		弁理士 崔海龍
審査請求日	令和3年7月9日(2021.7.9)	(72)発明者	徐異凌
(31)優先権主張番号	201710561850.2		中華人民共和国, 200240, 上海市閔行区東川路800号
(32)優先日	平成29年7月11日(2017.7.11)		
(33)優先権主張国・地域又は機関	中国(CN)		
(31)優先権主張番号	201710973473.3		
(32)優先日	平成29年10月18日(2017.10.18)		
	最終頁に続く		最終頁に続く

(54)【発明の名称】 メディア内容に基づく自己適応システムコードFEC符号化および復号化方法、装置、システムおよび媒体

(57)【特許請求の範囲】

【請求項1】

FEC符号化コードワードのデコード方法であって、  
送信された復元データパケット、中間データパケット及び対応する指示情報を受信し、前記中間データパケットは実際の応用シーンおよびメディア内容に基づいて分けられた異なる優先順位の高い方から低い方に再配列されたものであるステップと、  
前記復元データパケット及び中間データパケットに含まれるインデックスフィールド及び/又は優先順位指示フィールド、並びに前記指示情報を解析し、復元データパケット及び中間データパケットのロス状況、復元データパケット及び中間データパケットの前記インデックスフィールド、並びに前記指示情報に基づいて復元行列を生成するステップと、  
前記復元行列及び受信された復元データパケット、中間データパケットに基づいて中間コードワードを生成するステップと、  
中間コードワードを自己適応方式によりFEC復号化することで、復号化された後の中間データパケットに復元するステップと、  
復号化された後の中間データパケットにおけるサブパケットのヘッダの総情報を解析し、復号化された後の前記中間データパケットにおけるサブパケットを再配列することで、ソースデータパケットに復元するステップと、  
を含むことを特徴とする、FEC符号化コードワードのデコード方法。

【請求項2】

前記復元データパケットに含まれる前記インデックスフィールドは、前記復元データパ

ケットにおけるサブパケットのシーケンス、前記優先順位指示フィールドに対応する優先順位での復元データパケットにおけるサブパケットのシーケンス、及び／又は対応する優先順位に属するサブパケットの数目若しくは前記復元データパケットにおける前記サブパケットの割合を指示するものであることを特徴とする、請求項 1 に記載の F E C 符号化コードワードのデコード方法。

【請求項 3】

前記中間データパケットに含まれる前記インデックスフィールドは、前記中間データパケットにおけるサブパケットのシーケンス、及び／又は前記優先順位指示フィールドに対応する優先順位での中間データパケットにおけるサブパケットのシーケンスを指示するものであることを特徴とする、請求項 1 に記載の F E C 符号化コードワードのデコード方法。

10

【請求項 4】

前記指示情報は、前記復元データパケットの各優先順位でのサブパケット、及び／又は前記中間データパケットの各優先順位でのサブパケットを指示するものであることを特徴とする、請求項 1 に記載の F E C 符号化コードワードのデコード方法。

【請求項 5】

復元データパケット及び中間データパケットのロス状況、復元データパケット及び中間データパケットの前記インデックスフィールド、並びに前記指示情報に基づいて復元行列を生成するステップは、

復元データパケット、中間データパケットに含まれるインデックスフィールド及び／又は優先順位指示フィールド並びに前記指示情報に基づいて、中間データパケットのコーディング行列を生成するステップと、

20

中間データパケットのコーディング行列、復元データパケット、中間データパケットのロス状況に基づいて復元行列を生成するステップと、

を含むことを特徴とする、請求項 1 に記載の F E C 符号化コードワードのデコード方法。

【請求項 6】

中間コードワードを F E C 復号化することで復号化された後の中間データパケットに復元するステップは、

中間データパケットの L T コーディング行列及び中間コードワードに基づいて、復号化された後の中間データパケットを生成するステップを含むことを特徴とする、請求項 5 に記載の F E C 符号化コードワードのデコード方法。

30

【請求項 7】

送信された復元データパケット、中間データパケット及び対応する指示情報を受信し、前記中間データパケットは実際の応用シーンおよびメディア内容に基づいて分けられた異なる優先順位の高い方から低い方に再配列されたものである受信モジュールと、

前記復元データパケット及び中間データパケットに含まれるインデックスフィールド及び／又は優先順位指示フィールド、並びに前記指示情報を解析し、復元データパケット及び中間データパケットのロス状況、復元データパケット及び中間データパケットの前記インデックスフィールド、並びに前記指示情報に基づいて復元行列を生成する復元行列生成モジュールと、

前記復元行列及び受信した復元データパケット、中間データパケットに基づいて中間コードワードを生成し、中間コードワードを自己適応方式により F E C 復号化することで復号化された後の中間データパケットに復元する復号化モジュールと、

40

復号化された後の中間データパケットにおけるサブパケットのヘッダの総情報を解析し、復号化された後の前記中間データパケットにおけるサブパケットを再配列することで、ソースデータパケットに復元する復元モジュールと、

を含むことを特徴とする、F E C 符号化コードワードのデコード装置。

【請求項 8】

前記復元データパケットに含まれる前記インデックスフィールドは、前記復元データパケットにおけるサブパケットのシーケンス、復元データパケットにおけるサブパケットの前記優先順位指示フィールドに対応する優先順位でのシーケンス、及び／又は対応する優

50

先順位に属するサブパケットの数若しくは前記サブパケットの前記復元データパケットにおける割合を指示するものであることを特徴とする、請求項 7 に記載の F E C 符号化コードワードのデコード装置。

【請求項 9】

前記中間データパケットに含まれる前記インデックスフィールドは、前記中間データパケットにおけるサブパケットのシーケンス及び / 又は中間データパケットにおけるサブパケットの前記優先順位指示フィールドに対応する優先順位でのシーケンスを指示するものであることを特徴とする、請求項 7 に記載の F E C 符号化コードワードのデコード装置。

【請求項 10】

前記指示情報は、前記復元データパケットの各優先順位でのサブパケット、及び / 又は前記中間データパケットの各優先順位でのサブパケットを指示するものであることを特徴とする、請求項 7 に記載の F E C 符号化コードワードのデコード装置。

10

【請求項 11】

前記復元行列生成モジュールが復元データパケット及び中間データパケットのロス状況、復元データパケット及び中間データパケットの前記インデックスフィールド、並びに前記指示情報に基づいて復元行列を生成することは、

復元データパケット、中間データパケットに含まれるインデックスフィールド及び / 又は優先順位指示フィールド及び前記指示情報に基づいて中間データパケットのコーディング行列を生成すること、並びに

中間データパケットのコーディング行列及び復元データパケット、中間データパケットのロス状況に基づいて復元行列を生成すること、

20

を含むことを特徴とする、請求項 7 に記載の F E C 符号化コードワードのデコード装置。

【請求項 12】

中間データパケットの L T コーディング行列及び中間コードワードに基づいて復号化された後の中間データパケットを生成する復号化モジュールを含むことを特徴とする、請求項 11 に記載の F E C 符号化コードワードのデコード装置。

【請求項 13】

コンピュータプログラムが記憶されたコンピュータ読み取り可能な記憶媒体であって、

前記コンピュータプログラムは、プロセッサにより実行されることにより、請求項 1 から 6 のいずれか 1 項に記載の F E C 符号化コードワードのデコード方法を実現することを特徴とする、コンピュータ読み取り可能な記憶媒体。

30

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、中国特許出願 C N 2 0 1 7 1 0 9 7 3 4 7 3 . 3 ( 出願日 : 2 0 1 7 年 1 0 月 1 8 日、発明の名称 : メディア内容に基づく自己適応システムコード F E C 方法、装置およびシステム ) および中国特許出願 C N 2 0 1 7 1 0 5 6 1 8 5 0 . 2 ( 出願日 : 2 0 1 7 年 0 7 月 1 1 日、発明の名称 : メディア内容に基づく自己適応システムコード F E C 符号化および復号化方法 ) の優先権を主張する。

【0002】

40

本発明は、マルチメディア伝送技術分野に関し、具体的には、メディア内容に基づく自己適応システムコード F E C ( F o r w a r d E r r o r C o r r e c t i o n 、前方誤り訂正 ) 符号化および復号化方法、装置、システムおよび媒体に関する。

【背景技術】

【0003】

ヘテロジニアスネットワークのメディアサービスシステムにおいて、内容は、インターネットプロトコルまたは放送プロトコルにより端末に送信される。インターネットでは、I P / T C P または U D P メッセージによりメディアデータを伝送し、放送では、M P E G 2 - T S により内容を伝送する。U D P メッセージは、複数のネット機器を経過した後、ロスする恐れがある。放送 T S ストリームは、伝送環境の影響によりビットエラーが発

50

生し、端末側の画面の破損または音声の中断をもたらす恐れがある。

【0004】

FEC (Forward Error Correction、前方誤り訂正) 技術は、通信システムにおいて幅広く応用されている符号化技術である。サーバー側は、メディアデータに対して誤り訂正の符号化を行い、さらに冗長情報を追加して送信し、端末側は、逆方向のFEC復号化を行い、紛失されたメッセージについて復元を行う。典型的なブロックシンボルを例にすれば、その基本原理は以下のとおりである。

送信端において、 $k$  bitの情報を一つのブロックとして符号化し、 $(n - k)$  bitの冗長校正情報を追加して、長さが $n$  bitのコードワードを構成する。コードワードがチャネルを経て受信端に達した後、もし誤りが訂正可能な範囲内であれば、復号化によって誤りのbitを検査して訂正することができる。これにより、チャネルによりもたらした干渉に抵抗して、システムのビット誤り率を効果的に低減し、通信システムの信頼性を向上させる。しかしながら、FEC処理は冗長オーバーヘッドコストによりシステムのビット誤り率を低減するため、過剰なFEC符号化はシステムのリアルタイム性およびネットワークの状態に対してプレッシャーを与えることもある。

10

【0005】

不均一誤り保護 (Unequal Error Protection、UEP) は、ジョイントソースチャネルコーディングの一種である。その核心的な思想は、コードストリームの各部分のデータの異なる重要性に基づき、各部分のデータに対して異なるチャネル保護メカニズムを用い、即ち重要なコードストリームに対しては重点的に保護する。UEPは非重要なコードストリームのノイズに耐える性能を低下させるが、システムのビット誤りを防ぐ全体的な性能の向上には役立つ。

20

【0006】

FEC (前方誤り訂正) 符号化技術の一つとして、デジタルファウンテンコード (Digital Fountain Code) は伝送過程において、フィードバックおよび自動再送信のメカニズムを必要としないため、信号送受信による時間遅延およびブロードキャスト応用におけるフィードバック爆発の問題を回避している。デジタルファウンテンの基本的な思想は、次のとおりである。送信端は、オリジナルデータを $k$ 個のデータシンボルに分割し、これらのデータシンボルについて符号化し、任意の長さの符号化シンボルのコードストリームを出力する。受信端は、 $n$  ( $n$ は $k$ よりやや大きい) 個の符号化シンボルを正確に受信するだけで全ての $k$ 個のデータシンボルを大きな確率で復元することができる。数字ファウンテンコードそれ自体は、UEP性能を有するので、重要性が異なるデータの保護を実現することができる。

30

【発明の概要】

【0007】

従来技術の問題点に鑑みて、本発明は、メディア内容に基づく自己適応システムコードFEC符号化および復号化方法、装置、システムおよび媒体を提供することを目的とする。

【0008】

(1) 本発明に係るメディア内容に基づく自己適応システムコードFEC符号化および復号化方法は、

40

メディア内容の重要度に基づいて、ソースデータを優先順位に従って $N$  ( $1$ より大きい整数) 種類のソースデータパケットに分けるステップと、

ソースデータおよび $N$ 種類のソースデータパケットの優先順位に基づいて $N$ 種類の間中コードを生成するステップと、

前記 $N$ 種類の間中コードに基づいて、チャネル状況に従って $N$ 種類のソースデータの復元データを設定し、 $N$ 種類の符号化シンボルを生成するステップと、

前記符号化シンボルを受信し、復号化の需要に応じて前記符号化シンボルを整理、配列し、前記符号化シンボルを復号化処理するステップと、

受信した符号化シンボルの状況に基づいて、先に優先順位が高い中間コードを復号化し、この中間コードに基づいて対応するソースデータパケットに復元するステップと、

50

を含む。

【 0 0 0 9 】

( 2 ) 好ましくは、前記ソースデータおよびN種類のソースデータパケットの優先順位に基づいてN種類の間コードを生成するステップは、

N種類のソースデータパケットに対応するコーディング行列  $A_i$  (  $i = 1, 2, \dots, N$  ) を構成することを含み、

【数 1 】

$$A_i = \begin{bmatrix} G_{p_i} \\ G_{ENC_i} \end{bmatrix} \quad i = 1, 2, \dots, N$$

10

ここで、 $A_i$  は、第  $i$  のソースデータパケットに対応するコーディング行列、 $G_{p_i}$  は、第  $i$  のソースデータパケットに対応するLDPC行列および単位行列からなる結合行列、 $G_{ENC_i}$  は、第  $i$  のソースデータパケットに対応するLTコーディング行列を示し、N種類のソースデータパケットに対応するコーディング行列に基づいてソースデータのコーディング行列  $A$  を構築し、前記ソースデータのコーディング行列は、以下の通りであり、

【数 2 】

$$A = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{p_i} & \dots & 0 \\ 0 & G_{ENC_i} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_N} \\ 0 & 0 & \dots & G_{ENC_N} \end{bmatrix}$$

20

$$A \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} 0 \\ D_1 \\ 0 \\ \dots \\ D_i \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

30

したがって、

【数 3 】

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

40

であり、

ここで、 $C_1, C_2, \dots, C_i, \dots, C_N$  は、それぞれN種類のソースデータパケットの間コードに対応し、 $D_1, D_2, \dots, D_i, \dots, D_N$  は、それぞれN種類のソースデータパケットのデータに対応し、上付き - 1 は、逆行列演算子である。

【 0 0 1 0 】

50

( 3 ) 好ましくは、前記 N 種類の間コードに基づいて、チャンネル状況に応じて設定した N 種類のソースデータの復元データは、

【数 4】

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{ENC_N} \\ G_{ENC_{11}} & 0 & \dots & 0 \\ G_{ENC_{12}} & G_{ENC_{21}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC_{1N}} & G_{ENC_{2(N-1)}} & \dots & G_{ENC_{N1}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

10

を含み、

ここで、 $R_1, R_2, \dots, R_i, \dots, R_N$  は、それぞれ N 個のソースデータパケットの復元データに対応し、 $G_{ENC_{1i}}$  は、 $G_{ENC_1}$  と同じ生成ルールを有する LT コーディング行列を示し、 $G_{ENC_{i1}}$  は、 $G_{ENC_i}$  と同じ生成ルールを有する LT コーディング行列を示す。 $i = 1, 2, \dots, N$  である。

【0 0 1 1】

( 4 ) 好ましくは、N が 2 である場合、重要度が異なるデータに対して以下の対応する中間コードを生成することができ、

20

【数 5】

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \end{bmatrix}$$

LT コーディング行列の構造を変更することで  $G_{ENC_{11}}, G_{ENC_{12}}, G_{ENC_{21}}$  を得て、 $G_{ENC_{12}}$  は、 $G_{ENC_1}$  の生成方式と同じであり、 $G_{ENC_{21}}$  は、 $G_{ENC_2}$  の生成方式と同じであり、 $G_{ENC_{11}}$  は、 $G_{ENC_1}$  と同じ生成方式を有する LT コーディング行列を示し、 $G_{ENC_{21}}$  は、 $G_{ENC_2}$  と同じ生成方式を有する LT コーディング行列を示し、 $G_{ENC_{12}}$  は、 $G_{ENC_1}$  と同じ生成方式を有する LT コーディング行列を示し、

30

得られた復元データ  $R_1$  が優先順位 1 のデータにのみ関連し、復元データ  $R_2$  が優先順位 1 および 2 のデータに関連する場合、

【数 6】

$$\begin{bmatrix} D_1 \\ D_2 \\ R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 \\ 0 & G_{ENC_2} \\ G_{ENC_{11}} & 0 \\ G_{ENC_{12}} & G_{ENC_{21}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix}.$$

40

が得られる。

【0 0 1 2】

( 5 ) 好ましくは、N が 2 である場合、前記符号化シンボルを受信し、前記符号化シンボルを復号化処理するステップは、

第 1 段階の復号化の時に、復元行列 W および受信したソースデータパケットに基づいて中間コードワードを得て、

【数 7】

50

$$W \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ 0 \\ REV\_D_2 \\ REV\_R_1 \\ REV\_R_2 \end{bmatrix}$$

$$W = \begin{bmatrix} G\_p_1 & 0 \\ REV\_G\_ENC_1 & 0 \\ 0 & G\_p_2 \\ 0 & REV\_G\_ENC_2 \\ REV\_G\_ENC_{11} & 0 \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} \end{bmatrix} \quad 10$$

ここで、

$REV\_G\_ENC_1$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_1$  におけるロスしたパケットに対応する行を除去した後の  $LT$  コーディング行列を示し、

$REV\_G\_ENC_{11}$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_{11}$  におけるロスしたパケットに対応する行を除去した後の  $LT$  コーディング行列を示し、

$REV\_G\_ENC_{12}$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_{12}$  におけるロスしたパケットに対応する行を除去した後の  $LT$  コーディング行列を示し、

$REV\_G\_ENC_2$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_2$  におけるロスしたパケットに対応する行を除去した後の  $LT$  コーディング行列を示し、

$REV\_G\_ENC_{21}$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_{21}$  におけるロスしたパケットに対応する行を除去した後の  $LT$  コーディング行列を示し、

【数 8】

$$REV\_A = \begin{bmatrix} G\_p_1 & 0 \\ REV\_G\_ENC_1 & 0 \\ REV\_G\_ENC_{11} & 0 \\ 0 & G\_p_2 \\ 0 & REV\_G\_ENC_2 \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} \end{bmatrix};$$

にし、受信した対応優先順位でのソースデータパケットの数および受信したデータパケットの総数に基づいてデコードし、

受信した符号化シンボルの総数がソースデータで生成した符号化シンボルの数よりも少なく、かつ、

【数 9】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_{11} \end{bmatrix}$$

の行数が列数より少ない場合、 $C_1$ 、 $C_2$  はいずれも算出できず、

受信した符号化シンボルの総数がソースデータで生成した符号化シンボルの数よりも少なく、かつ、

【数 1 0】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_{11} \end{bmatrix}$$

の行数が列数以上である場合、

【数 1 1】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_{11} \end{bmatrix}$$

10

の行数が列数以上であると、 $REV\_A_1$  はリバーシブルであり、対応する逆行列は  $REV\_A_1^{-1}$  と記し、

【数 1 2】

$$[C_1] = REV\_A^{-1} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \end{bmatrix}$$

20

が得られ、ここで、 $C_2$  には解がなく、

ここで、 $REV\_D_1$  は、受信した優先順位 1 のソースデータ、 $REV\_R_1$  は、受信した優先順位 1 のソースデータの復元データを示し、

受信した符号化シンボルの総数がソースデータで生成した符号化シンボルの数以上であり、かつ、

【数 1 3】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_{11} \end{bmatrix}$$

30

の行数が列数以上である場合、

【数 1 4】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_{11} \end{bmatrix}$$

の行数が列数以上であると、 $REV\_A_1$  はリバーシブルであり、対応する逆行列は  $REV\_A_1^{-1}$  と記し、

40

【数 1 5】

$$REV\_A_2 = \begin{bmatrix} G\_p_2 \\ REV\_G\_ENC_2 \\ REV\_G\_ENC_{21} \end{bmatrix}$$

の行数が列数以上であると、 $REV\_A_2$  はリバーシブルであり、対応する逆行列は  $REV\_A_2^{-1}$  と記し、 $REV\_A_1$  および  $REV\_A_2$  がいずれもリバーシブルであると、 $REV\_A$  はリバーシブルであり、対応する逆行列は  $REV\_A^{-1}$  と記し、

50



【数 1 6】

$$B_{1,1} = \begin{bmatrix} 0 \\ 0 \\ \text{REV\_G\_ENC}_{12} \end{bmatrix}$$

であると、

【数 1 7】

$$\text{REV\_A}^{-1} = \begin{bmatrix} \text{REV\_A}_1^{-1} & 0 \\ \text{REV\_A}_2^{-1}B_{1,1}\text{REV\_A}_1^{-1} & \text{REV\_A}_2^{-1} \end{bmatrix} \quad 10$$

であり、

【数 1 8】

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \text{REV\_A}^{-1} \begin{bmatrix} 0 \\ \text{REV\_D}_1 \\ \text{REV\_R}_1 \\ 0 \\ \text{REV\_D}_2 \\ \text{REV\_R}_2 \end{bmatrix} = \begin{bmatrix} \text{REV\_A}_1^{-1} & 0 \\ \text{REV\_A}_2^{-1}B_{1,1}\text{REV\_A}_1^{-1} & \text{REV\_A}_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ \text{REV\_D}_1 \\ \text{REV\_R}_1 \\ 0 \\ \text{REV\_D}_2 \\ \text{REV\_R}_2 \end{bmatrix}; \quad 20$$

が得られ、

【数 1 9】

$$\text{REV\_A}_2 = \begin{bmatrix} G_{p_2} \\ \text{REV\_G\_ENC}_2 \\ \text{REV\_G\_ENC}_{21} \end{bmatrix}$$

の行数が列数よりも少ないと、REV\_A<sub>2</sub>には逆行列が存在せず、

【数 2 0】

30

$$[C_1] = \text{REV\_A}^{-1} \begin{bmatrix} 0 \\ \text{REV\_D}_1 \\ \text{REV\_R}_1 \end{bmatrix};$$

しか算出できず、

ここで、REV\_D<sub>2</sub>は、受信した優先順位2のソースデータ、REV\_R<sub>2</sub>は、受信した優先順位1および2のソースデータの復元データを示し、

受信した符号化シンボルの総数がソースデータで生成した符号化シンボルの数以上であり、かつ、

【数 2 1】

40

$$\text{REV\_A}_1 = \begin{bmatrix} G_{p_1} \\ \text{REV\_G\_ENC}_1 \\ \text{REV\_G\_ENC}_{11} \end{bmatrix}$$

の行数が列数よりも少なく、

【数 2 2】

$$REV\_A_2 = \begin{bmatrix} G\_p_2 \\ REV\_G\_ENC_2 \\ REV\_G\_ENC_{21} \end{bmatrix}$$

の行数が列数以上である場合、ガウスの消去法により行列  $REV\_A$  を直接解く、即ち、以下の線型方程式系を解くことにより、 $C_1$ 、 $C_2$  が得られる。

【数 2 3】

$$\begin{bmatrix} G\_p_1 & 0 \\ REV\_G\_ENC_1 & 0 \\ REV\_G\_ENC_{11} & 0 \\ 0 & G\_p_2 \\ 0 & REV\_G\_ENC_2 \\ \_REV\_G\_ENC_{12} & REV\_G\_ENC_{21} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \end{bmatrix} \quad 10$$

【 0 0 1 3】

( 6 ) 好ましくは、一部のデータがロスしたので、受信した符号化シンボルに基づいて復元行列を生成する必要がある、つまり、ロスした符号化シンボルに対応する行列行を消去し、対応する復元行列 ( $REV\_G\_ENC$  と記す) を生成し、受信したデータパケットの数に応じてケースバイケースで中間コードワードを生成し、具体的には、以下のステップ A 1 ~ A 4 を含み、

( ステップ A 1 )

1 個の優先順位のデータパケットを受信した場合、以下の復元行列方程式が得られ、

【数 2 4】

$$\begin{bmatrix} G\_p_1 & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ 0 & G\_p_2 & \dots & 0 \\ 0 & REV\_G\_ENC_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G\_p_l \\ 0 & 0 & \dots & REV\_G\_ENC_l \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ REV\_G\_ENC_{1l} & REV\_G\_ENC_{2(l-1)} & \dots & REV\_G\_ENC_{l1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ 0 \\ REV\_D_2 \\ \dots \\ 0 \\ REV\_D_l \\ REV\_R_1 \\ REV\_R_2 \\ \dots \\ REV\_R_l \end{bmatrix} \quad 30$$

復元行列方程式に対して一次行変換を行うことで、

【数 2 5】

$$\begin{bmatrix} G\_p_1 & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ 0 & G\_p_2 & \dots & 0 \\ 0 & REV\_G\_ENC_2 & \dots & 0 \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G\_p_l \\ 0 & 0 & \dots & REV\_G\_ENC_l \\ REV\_G\_ENC_{1l} & REV\_G\_ENC_{2(l-1)} & \dots & REV\_G\_ENC_{l1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \\ \dots \\ 0 \\ REV\_D_l \\ REV\_R_l \end{bmatrix} \quad 40$$

が得られ、

ここで、

$REV\_G\_ENC_1$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC$

$1$  におけるロスしたパケットに対応する行を除去した後の  $L T$  コーディング行列を示し、  
 $REV\_G\_ENC_{11}$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_{11}$  におけるロスしたパケットに対応する行を除去した後の  $L T$  コーディング行列を示し、

$REV\_G\_ENC_{12}$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_{12}$  におけるロスしたパケットに対応する行を除去した後の  $L T$  コーディング行列を示し、

$REV\_G\_ENC_2$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_2$  におけるロスしたパケットに対応する行を除去した後の  $L T$  コーディング行列を示し、

$REV\_G\_ENC_{21}$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_{21}$  におけるロスしたパケットに対応する行を除去した後の  $L T$  コーディング行列を示し、

$REV\_D_1$  は、受信した優先順位  $1$  のソースデータ、 $REV\_R_1$  は、受信した優先順位  $1$  のソースデータの復元データ、 $REV\_D_2$  は、受信した優先順位  $2$  のソースデータ、 $REV\_R_2$  は、受信した優先順位  $1$  および  $2$  ソースデータの復元データを示し、

$REV\_G\_ENC_{11}$  は、受信した符号化シンボルのパケットロス状況に応じて  $G\_ENC_{11}$  におけるロスしたパケットに対応する行を除去した後の  $L T$  コーディング行列を示し、

$REV\_D_1$  は、受信した優先順位  $1$  のソースデータ、 $REV\_R_1$  は、受信した優先順位  $1$  から  $1$  のソースデータの復元データ、 $C_1$  は、生成した第  $1$  の中間コードを示し、

(ステップ  $A2$ )

【数 2 6】

$$REV\_A = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV\_G\_ENC_2 \dots & \dots & \dots \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_l} \\ 0 & 0 & \dots & REV\_G\_ENC_l \\ REV\_G\_ENC_{1l} & REV\_G\_ENC_{2(l-1)} & \dots & REV\_G\_ENC_{ll} \end{bmatrix}$$

とし、

$R(i)$  を受信して拡張した後の第  $i$  ( $1 \leq i \leq l$ ) の符号化シンボルの数、 $L(i)$  を符号化端の各レベルのシンボルの長さとし、各レベルで生成した行列のランクを計算し、  
 $eachRank(i) = \min(Len(R(i)), sum(L(1:i)))$

式中： $eachRank(i)$  は、優先順位  $i$  のデータに対応するサブ行列のランクを示し、ここで、 $i = 2$  である場合、対応するサブ行列は、

【数 2 7】

$$\begin{bmatrix} 0 & G_{p_2} \\ 0 & REV\_G\_ENC_2 \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} \end{bmatrix}$$

であり、 $Len(R(i))$  は、受信した優先順位  $i$  の符号化シンボルの数の長さを示し、 $sum(L(1:i))$  は、優先順位  $i$  のサブ行列の列数を示し、 $\min()$  は、最小値演算を示し、

累積行列のランクを計算し、

$$cumRank(i) = \min(cumRank(i-1) + eachRank(i), sum(L(1:i)))$$

ここで、 $cumRank(i)$  は、優先順位  $1$  から優先順位  $i$  に対応するサブ行列ラン

クの和を示す。cumRank(i-1)は、優先順位1から優先順位i-1に対応するサブ行列ランクの和を示し、

$$\text{cumRank}(1) = \text{eachRank}(1)$$

順にcumRank(i) ≥ sum(L(1:i))を判断し、成立した場合、cumFullRank(i) = 1であり、成立しない場合、cumFullRank(i) = 0であり、cumFullRank(i)は、優先順位1から優先順位iに対応するサブ行列からなる行列のフルランクを示し、

(ステップA3)

受信した各レベルの符号化シンボルの数に応じて、ケースバイケースでそれぞれ中間コードC<sub>1</sub>、C<sub>2</sub>...C<sub>1</sub>を求める。ここで、i初期値を1に設定し、

cumFullRank(i) == 1 && cumFullRank(1:i) > 0を満足した場合、RFC 6330復号化方法によりC<sub>1</sub> ~ C<sub>i</sub>を解き、ここで、&&は、AND演算を示し、cumFullRank(1:i)は、優先順位1から優先順位iの各サブ行列からなる行列のフルランクを示し、C<sub>1</sub>は、優先順位1のデータに対応する中間コードワードを示し、C<sub>i</sub>は、優先順位iのデータに対応する中間コードを示し、

【数28】

$$\text{REV}_{A_{\Delta i}} = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ \text{REV}_{G\_ENC_1} & 0 & \dots & 0 \\ \text{REV}_{G\_ENC_{11}} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & \text{REV}_{G\_ENC_2} \dots & \dots & \dots \\ \text{REV}_{G\_ENC_{12}} & \text{REV}_{G\_ENC_{21}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & \text{REV}_{G\_ENC_i} \\ \text{REV}_{G\_ENC_{1i}} & \text{REV}_{G\_ENC_{2(i-1)}} & \dots & \text{REV}_{G\_ENC_{i1}} \end{bmatrix}$$

とし、

【数29】

$$\text{REV}_{A_{\Delta i}}^{-1} = \begin{bmatrix} A_1^{-1} & 0 & 0 & 0 \\ X_{1,2} & A_2^{-1} & 0 & 0 \\ \dots & \dots & \dots & 0 \\ X_{1,i} & X_{2,(i-1)} & \dots & A_i^{-1} \end{bmatrix}$$

が得られ、

ここで、

【数30】

$$X_{1,2} = A_2^{-1} B_{1,2} A_1^{-1}, \quad X_{a,b} = A_b^{-1} B_{a,b} A_a^{-1} + \sum_{k=a+1}^{b-1} A_b^{-1} B_{k,b} X_{a,k},$$

であり、

ここで、REV\_A<sub>i</sub>は、優先順位がiである場合の全てのサブ行列からなるフルランク行列を示し、B<sub>1,2</sub>は、サブ行列

【数31】

$$\begin{bmatrix} 0 \\ 0 \\ \text{REV}_{G\_ENC_{12}} \end{bmatrix}$$

10

20

30

40

50

を示し、 $X_{1\_2}$ は、サブ行列 $B_{1\_2}$ を逆演算した後のサブ行列を示し、 $X_{a\_b}$ は、サブ行列 $B_{a\_b}$ を逆演算して得られたサブ行列を示し、 $B_{a\_b}$ は、サブ行列  
【数 3 2】

$$\begin{bmatrix} 0 \\ 0 \\ REV\_G\_ENC_{ab} \end{bmatrix}$$

を示し、 $a$ の範囲は $1 \sim i$ であり、 $b$ の範囲は $2 \sim i$ であり、 $A_i^{-1}$ は、サブ行列  
【数 3 3】

10

$$\begin{bmatrix} G_{p_i} \\ REV\_G\_ENC_i \\ REV\_G\_ENC_{i1} \end{bmatrix}$$

の逆行列を示し、  
【数 3 4】

$$C_{i\_temp} = REV\_A_i^{-1} \begin{bmatrix} 0 \\ REV\_D_i \\ REV\_R_i \end{bmatrix}$$

20

とし、ここで、  
【数 3 5】

$$REV\_A_i = \begin{bmatrix} G_{p_i} \\ REV\_G\_ENC_i \\ REV\_G\_ENC_{i1} \end{bmatrix}$$

であり、 $REV\_A_i^{-1}$ は、 $REV\_A_i$ の逆行列である場合、  
【数 3 6】

30

$$\begin{aligned} C_1 &= C_{1\_temp} \\ C_2 &= C_{2\_temp} + X_{1\_2} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \end{bmatrix} = C_{2\_temp} + A_2^{-1} B_{1\_2} C_1 \\ C_3 &= C_{3\_temp} + X_{1\_3} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \end{bmatrix} + X_{2\_3} \begin{bmatrix} 0 \\ REV\_D_2 \\ REV\_R_2 \end{bmatrix} = C_{3\_temp} + A_3^{-1} B_{1\_3} C_1 + A_3^{-1} B_{2\_3} C_2 \end{aligned}$$

40

が得られ、  
数学的帰納法により、  
【数 3 7】

$$C_i = C_{i\_temp} + \sum_{j=1}^{i-1} X_{j\_i} \begin{bmatrix} 0 \\ REV\_D_j \\ REV\_R_j \end{bmatrix} = C_{i\_temp} + \sum_{j=1}^{i-1} A_i^{-1} B_{j\_i} C_j$$

が得られ、 $C_1$ 、 $C_2 \dots C_i$ が算出され、 $C_{i+1} \dots C_L$ が算出できず、  
 $cumFullRand(i) = 1$ を満足し、 $cumFullRank(1:i) >$

50

0 を満足しない場合、ガウス消去復号化法により  $C_1 \sim C_i$  を解き、  
【数 3 8】

$$REV_{A_{\Delta i}} = \begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV\_G\_ENC_2 \dots & \dots & \dots \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & REV\_G\_ENC_i \\ REV\_G\_ENC_{1i} & REV\_G\_ENC_{2(i-1)} & \dots & REV\_G\_ENC_{i1} \end{bmatrix} \quad 10$$

とし、ガウスの消去法により解線型方程式系：

【数 3 9】

$$\begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & REV\_G\_ENC_2 \dots & \dots & \dots \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_i} \\ 0 & 0 & \dots & REV\_G\_ENC_i \\ REV\_G\_ENC_{1i} & REV\_G\_ENC_{2(i-1)} & \dots & REV\_G\_ENC_{i1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \\ \dots \\ 0 \\ REV\_D_i \\ REV\_i \end{bmatrix} \quad 20$$

を解くことにより、 $C_1$ 、 $C_2 \dots C_i$  を算出できるが、 $C_{i+1} \dots C_L$  を算出できず、  
任意の  $i$  値に対して、 $cumFullRank(i) = 1$  を満足しない場合、対応する  $C_i$  を算出できず、 $i = 1$  まで  $cumFullRank(1:i) > 0$  も満足できない場合、 $C_1$ 、 $C_2 \dots C_L$  はいずれも算出できず、復号化が失敗し、

(ステップ A 4)

$i$  の値を 1 減らし、ステップ A 3 に戻る。

【0 0 1 4】

(7) 好ましくは、ソースデータパケットにおける各サブパケットのヘッダの重要な情報を解析し、

前記重要な情報に基づいて、前記ソースデータパケットにおける各サブパケットをそれぞれ異なる優先順位にマッピングし、

サブパケットの優先順位に従って、サブパケットを再配列し、中間データパケットを得て、

前記中間データパケットを複数のシンボル  $symbol$  に変換し、複数のシンボル  $symbol$  に変換された中間データパケットに対して前方誤り訂正 FEC 符号化を行なった後、複数のシンボル  $symbol$  を含む復元データパケットを得て、

前記復元データパケットにインデックスフィールドおよび/または優先順位指示フィールドを追加し、前記インデックスフィールドは、復元データパケットにおけるサブパケットのシーケンス、復元データパケットにおけるサブパケットの前記優先順位指示フィールドに対応する優先順位でのシーケンスを指示し、前記中間データパケットにもインデックスフィールドおよび/または優先順位指示フィールドを追加し、前記インデックスフィールドは、中間データパケットにおけるサブパケットのシーケンスおよび/または中間データパケットにおけるサブパケットの前記優先順位指示フィールドに対応する優先順位でのシーケンスを指示し、

前記復元データパケットおよび中間データパケットに対応する指示情報を追加し、また

30

40

50

は復元データパケットおよび中間データパケットを送信するときに対応する指示情報を送信し、前記指示情報は、復元データパケットの各優先順位でのサブパケット、対応する優先順位に属するサブパケットの数または前記サブパケットの前記復元データパケットにおける割合を示し、前記指示情報は、中間データパケットの各優先順位でのサブパケット、および対応する優先順位に属するサブパケットの数または前記サブパケットの前記中間データパケットにおける割合を示し、

端末側に指示情報を含む復元データパケット、中間データパケットを送信し、または端末側に復元データパケット、中間データパケットおよび対応する指示情報を送信する、ステップをさらに含む。

【 0 0 1 5 】

10

( 8 ) 好ましくは、ソースデータパケットにおける各サブパケットのヘッダの重要な情報を解析することは、

各サブパケットのヘッダにおける *p r i o r i t y* フィールドを読み取り、前記 *p r i o r i t y* フィールドにおける情報に基づいて対応するサブパケットの優先順位を確定することを含む。

【 0 0 1 6 】

( 9 ) 好ましくは、重要な情報に基づいて前記ソースデータパケットにおけるサブパケットをそれぞれ異なる所定の優先順位にマッピングすることは、

前記ソースデータパケットで実際に使用される優先順位の数および各優先順位での前記 *p r i o r i t y* フィールドとのマッピング関係を確定すること、および

20

前記マッピング関係に基づいて、サブパケットを対応する所定の優先順位にマッピングすることを含む。

【 0 0 1 7 】

( 1 0 ) 好ましくは、前記中間データパケットを複数のシンボル *s y m b o l* に変換し、複数のシンボル *s y m b o l* に変換された中間データパケットに対して前方誤り訂正 F E C 符号化を行なった後、複数の *s y m b o l* を含む復元データパケットを得ることは、

前記中間データパケットに対して、R a p t o r Q システムコードの自己適応 F E C 符号化方式により F E C 符号化を行い、以下の中間コードワードを得ること、および

【数 4 0 】

30

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \vdots \\ 0 \\ D_N \end{bmatrix}$$

ここで、 $A_i$  は、第  $i$  のソースデータパケットに対応するコーディング行列を示し、 $C_1$  ,  $C_2$  ,  $\dots$  ,  $C_i$  ... ,  $C_N$  は、それぞれ  $N$  種類のソースデータパケットの中間コードに対応し、 $D_1$  ,  $D_2$  ,  $\dots$  ,  $D_i$  ... ,  $D_N$  は、それぞれ  $N$  種類のソースデータパケットのデータに対応し、上付き - 1 は、逆行列演算子であり、

40

前記中間コードワードに基づいて復元データパケットを取得し、前記復元データパケットに F E C *p a y l o a d I D* を追加することを含み、ここで、前記復元データパケットにおける復元データは以下の通りであり、

【数 4 1 】

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENC_N} \\ G_{ENC_{11}} & 0 & \dots & 0 \\ G_{ENC_{12}} & G_{ENC_{21}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC_{1N}} & G_{ENC_{2(N-1)}} & \dots & G_{ENC_{N1}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

ここで、 $R_1, R_2, \dots, R_i, \dots, R_N$ は、それぞれ1～N個のソースデータパケットの復元データに対応し、 $G_{ENC_i}$ は、第iのソースデータパケットに対応するLTコーディング行列を示し、 $G_{ENC_{1i}}$ は、 $G_{ENC_1}$ と同じ生成ルールを有するLTコーディング行列を示し、 $G_{ENC_{i1}}$ は、 $G_{ENC_i}$ と同じ生成ルールを有するLTコーディング行列を示し、 $i = 1, 2, \dots, N$ である。

【0018】

(11) 本発明に係るメディア内容に基づく自己適応システムコードFEC符号化および復号化方法は、

サーバーが送信した復元データパケット、中間データパケットを受信し、または復元データパケット、中間データおよび対応する指示情報を受信し、

前記復元データパケットおよび中間データパケットに含まれるインデックスフィールドを解析し、前記インデックスフィールドに基づいてコーディング行列を生成し、前記インデックスフィールドは、復元データパケットにおけるサブパケットのシーケンス、および中間データパケットの各優先順位でのサブパケットを指示し、復元データパケットおよび中間データパケットの優先順位指示フィールドによりインデック情報を直接取得し、前記インデック情報に基づいてコーディング行列を生成し、前記インデック情報は、復元データパケットにおけるサブパケットのシーケンス、および中間データパケットの各優先順位でのサブパケットを指示し、

前記コーディング行列に対して前方誤り訂正FEC復号化を行い、中間データパケットに復元し、

前記中間データパケットのヘッダにおける情報を解析し、前記中間データパケットのサブパケット的シーケンスを再配列し、ソースデータパケットに復元する、

とのステップを含む。

【0019】

(12) 好ましくは、前記復元データパケットおよび中間データパケットに含まれるインデックスフィールドを解析することは、

以下の一般式によりソースデータに対応するコーディング行列の対応行インデックスフィールドを得ることを含み、

$\text{index} = \text{FEC payload ID} -$

$\text{number of packets of all priorities before this packet}$

ここで、 $\text{index}$ は、ソースデータに対応するコーディング行列の対応行インデックを示し、 $\text{FEC payload ID}$ は、データパケットのヘッダのインデック情報を示し、 $\text{number of packets of all priorities before this packet}$ は、このデータパケットの前の全ての優先順位のデータパケット総数を示す。

【0020】

(13) 好ましくは、前記コーディング行列に対して前方誤り訂正FEC復号化を行い、中間データパケットに復元することは、

前記コーディング行列を一次行変換し、受信した対応復元データパケットにおけるシンボル $\text{symbol}$ を配列し、復元されたデータパケットのヘッダのインデック情報 $\text{FEC payload\_ID}$ の値および対応優先順位でのシンボル $\text{symbol}$ の数に基づいてこ

10

20

30

40

50



のシンボル `symbol` が属する優先順位および変換後の位置を確定すること、或いは復元データパケットのヘッダにおける優先順位フィールドおよびこの優先順位フィールドに対応する優先順位でのインデック情報に基づいて、シンボル `symbol` が属する優先順位および変換後の位置を確定すること、

復元データパケットにおけるシンボル `symbol` のシーケンスを調整し、ガウス消去デコードおよび `RFC6330` デコードにより復号化することにより、復元された複数の `symbol` を含む中間データパケットを得ること、および

複数の `symbol` を含む中間データパケットを中間データパケットに変換すること、を含む。

【0021】

(14) 本発明に係るメディア内容に基づく自己適応システムコード `FEC` 符号化および復号化装置は、

ソースデータパケットにおける各サブパケットのヘッダの重要な情報を解析する解析モジュールと、

前記重要な情報に基づいて、前記ソースデータパケットにおける各サブパケットをそれぞれ異なる所定優先順位にマッピングするマッピングモジュールと、

所定優先順位に基づいてサブパケットを再配置することにより中間データパケットを得る配列モジュールと、

前記中間データパケットを複数のシンボル `symbol` に変換し、複数の `symbol` に変換された中間データパケットに対して前方誤り訂正 `FEC` 符号化を行うことで、複数の `symbol` を含む復元データパケットを得る符号化モジュールと、

前記復元データパケットにインデックスフィールドおよび/または優先順位指示フィールドを加え、前記インデックスフィールドは、復元データパケットにおけるサブパケットのシーケンスおよび/または復元データパケットにおけるサブパケットの前記優先順位指示フィールドに対応する優先順位でのシーケンスを指示するものであり、前記中間データパケットにインデックスフィールドおよび/または優先順位指示フィールドを加え、前記インデックスフィールドは、中間データパケットにおけるサブパケットのシーケンス、および/または中間データパケットにおけるサブパケットの前記優先順位指示フィールドに対応する優先順位でのシーケンスを指示するものであるインデックモジュールと、

前記復元データパケットおよび中間データパケットに対応する指示情報を加え、または復元データパケットおよび中間データパケットを送信するときに対応する指示情報を送信する指示情報生成モジュールであって、前記指示情報は、復元データパケットの各優先順位でのサブパケット、および対応する優先順位に属するサブパケットの数または前記サブパケットの前記復元データパケットにおける割合を指示し、前記指示情報は、中間データパケットの各優先順位でのサブパケット、および対応する優先順位に属するサブパケットの数または前記サブパケットの前記中間データパケットにおける割合を指示する指示情報生成モジュールと、

端末側に前記復元データパケット、中間データパケットおよび対応するシグナリング情報を送信し、または端末側に復元データパケット、中間データパケットおよび対応する指示情報を送信する送信モジュールと、

を含む。

【0022】

(15) 好ましくは、前記符号化モジュールは、具体的には、

前記中間データパケットに対して `RaptorQ` システムコードの自己適応 `FEC` 符号化方式により `FEC` 符号化を行うことで、以下の中間コードワードを取得し、

【数42】

10

20

30

40

50

$$\begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \vdots \\ 0 \\ D_N \end{bmatrix}$$

ここで、 $A_i$ は、第 $i$ のソースデータパケットに対応するコーディング行列を示し、 $C_1, C_2, \dots, C_i, \dots, C_N$ は、それぞれ $N$ 種類のソースデータパケットの中間コードに対応し、 $D_1, D_2, \dots, D_i, \dots, D_N$ は、それぞれ $N$ 種類のソースデータパケットのデータに対応し、上付き $-1$ は、逆行列演算子であり、

前記中間コードワードに基づいて復元データパケットを取得し、前記復元データパケットにFEC payload IDを追加し、ここで、前記復元データパケットにおける復元データは、次の通りであり、

【数43】

$$\begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_N \\ R_1 \\ R_2 \\ \vdots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_N \end{bmatrix}$$

ここで、 $R_1, R_2, \dots, R_i, \dots, R_N$ はそれぞれ $1 \sim N$ 個のソースデータパケットの復元データに対応し、 $G_{ENCi}$ は、第 $i$ のソースデータパケットに対応するLTコーディング行列を示し、 $G_{ENC1i}$ は、 $G_{ENC1}$ と同じ生成ルールを有するLTコーディング行列を示し、 $G_{ENCi1}$ は、 $G_{ENCi}$ と同じ生成ルールを有するLTコーディング行列を示し、 $i = 1, 2, \dots, N$ である。

【0023】

(16) 本発明に係るメディア内容に基づく自己適応システムコードFEC符号化および復号化装置は、受信モジュールと、コーディング行列生成モジュールと、復号化モジュールと、復元モジュールとを含み、

受信モジュールは、サーバーが送信した復元データパケット、中間データパケットを受信し、または復元データパケット、中間データおよび対応する指示情報を受信し、

コーディング行列生成モジュールは、前記復元データパケットおよび中間データパケットに含まれるインデックスフィールドを解析し、前記インデックスフィールドに基づいてコーディング行列を生成し、前記インデックスフィールドが復元データパケットにおけるサブパケットのシーケンス、および中間データパケットの各優先順位でのサブパケットを指示し、或いは

復元データパケットおよび中間データパケットの優先順位指示フィールドによりインデック情報を直接取得し、前記インデック情報に基づいてコーディング行列を生成し、前記インデック情報が復元データパケットにおけるサブパケットのシーケンス、および中間データパケットの各優先順位でのサブパケットを指示し、

復号化モジュールは、前記コーディング行列に対して前方誤り訂正FEC復号化を行い、中間データパケットに復元し、

復元モジュールは、前記中間データパケットのヘッダにおける情報を解析し、前記中間データパケットのサブパケットのシーケンスを再配列し、ソースデータパケットに復元する。

10

20

30

40

50

## 【0024】

(17) 好ましくは、前記コーディング行列生成モジュールは、具体的には、以下の一般式によりソースデータに対応するコーディング行列の対応行のインデックスフィールドを得て、  
index=FEC payload ID-  
number of packets of all priorities before this packet

ここで、indexは、ソースデータに対応するコーディング行列の対応行インデックを示し、FEC payload IDは、データパケットのヘッダのインデック情報を示し、number of packets of all priorities before this packetは、このデータパケットの前の全ての優先順位のデータパケット総数を示す。

10

## 【0025】

(18) 好ましくは、前記復号化モジュールは、具体的には、前記コーディング行列を一次行変換し、受信した対応復元データパケットにおけるシンボルsymbolを配列し、復元されたデータパケットのヘッダのインデック情報FEC payload IDの値および対応優先順位でのシンボルsymbolの数に基づいてこのシンボルsymbolが属する優先順位および変換後の位置を確定し、或いは復元データパケットのヘッダにおける優先順位フィールドおよびこの優先順位フィールドに対応する優先順位でのインデック情報に基づいて、シンボルsymbolが属する優先順位および変換後の位置を確定し、

20

復元データパケットにおけるシンボルsymbolのシーケンスを調整し、ガウス消去デコーダおよびRFC6330デコーダにより復号化することにより、復元された複数のsymbolを含む中間データパケットを取得し、

複数のsymbolを含む中間データパケットを中間データパケットに変換する。

## 【0026】

(19) 本発明に係るメディア内容に基づく自己適応システムコードFEC符号化および復号化システムは、サーバーおよび少なくとも1つの端末を含み、前記サーバーは、上記(7)~(10)のいずれか1つに記載のメディア内容に基づく自己適応システムコードFEC符号化および復号化方法を実行し、前記端末は、上記(11)~(13)のいずれか1つに記載のメディア内容に基づく自己適応システムコードFEC符号化および復号化方法を実行する。

30

## 【0027】

(20) 本発明は、コンピュータプログラムが記憶されたコンピュータ読み取り可能な記憶媒体をさらに提供する。前記コンピュータプログラムは、プロセッサにより実行されることにより、上記(7)~(13)のいずれか1つに記載の画像に基づくストライプ生地の検出位置決めクロッピング方法のステップを実行する。

## 【0028】

従来技術に比べて、本発明は、以下の有益な効果を有する。

本発明に係るメディア内容に基づく自己適応システムコードFEC方法は、メディア内容の重要度に応じて優先順位に従って配列し、不均一誤り保護(UEP)を採用することにより、メディア内容の品質が最大限に保証されるとともに、FECによるデータ冗長が減少される。また、システムコードの符号化および復号化がより柔軟になり、ソースデータストリームをマルチストリーム化する必要がないため、送信端FEC符号化の複雑さが低減され、FEC符号化の効率が向上される。さらに、現在のネットワーク状態の変化に基づいて符号化スキームを調整することができるため、時変ネットワークに対してより強い適応性を有する。

40

## 【図面の簡単な説明】

## 【0029】

以下の図面により非制限的な実施例を詳しく説明することにより、本発明の他の特徴、目的および利点は、より明確になる。

50

【図 1】メディアソースデータに対する F E C 二層構造の模式図である。  
【図 2】画像群における各フレームの依存関係の模式図である。  
【図 3】画像群における各フレームの依存関係の模式図である。  
【図 4】ファウンテンコードの不均一誤り保護符号化スキームの模式図である。  
【図 5】ファウンテンコードの不均一誤り保護のシステム構成の模式図である。  
【図 6】システム R a p t o r Q コードのコーディング行列の構造模式図である。  
【図 7】システム R a p t o r Q コードの不均一誤り保護のコーディング行列の構造模式図である。

【図 8】L T コーディング行列の不均一誤り保護構造の模式図である。  
【図 9】複数の優先順位の不均一誤り保護システムファウンテンコード符号化のフローチャートである。  
【図 10】複数の優先順位の不均一誤り保護システムファウンテンコード復号化のフローチャートである。  
【図 11】メディア内容に基づく自己適応システムコード F E C システムのフレーム図である。

10

【図 12】各優先順位のマッピング関係の模式図である。  
【図 13】優先順位に従ってデータパケットストリームを再配列する模式図である。  
【図 14】F E C 符号化後のソースデータパケットおよび復元データパケットに対する処理フローチャートである。  
【図 15】受信端がシグナリング情報およびデータパケット情報を解析する模式図である。  
【図 16】受信したデータパケットに基づいてコーディング行列を生成する模式図である。  
【図 17】受信したデータパケットを処理するフローチャートである。  
【図 18】復元データパケットストリームの初期シーケンスの結果の模式図である。  
【図 19】受信端がデータパケットのヘッダ情報を解析する模式図である。

20

【発明を実施するための形態】

【0030】

以下、具体的な実施形態を例示しながら、本発明について詳細に説明する。以下の実施形態は、いわゆる当業者が本発明をより理解しやすくするための例示であり、いかなる形式で本発明を限定するものではない。また、いわゆる当業者であれば、本発明の要旨から逸脱しない範囲内において、様々な変形や改良を行うことができ、これらも本発明の保護範囲に属することには注意されたい。

30

【0031】

以下、本明細書で用いられる用語を解釈する。

1) 不均一誤り保護 ( Unequal Error Protection、UEP ) は、ジョイントソースチャネルコーディングの一種である。その核心的な思想は、コードストリームの各部分のデータの異なる重要性に基づき、各部分のデータに対して異なるチャネル保護メカニズムを用い、即ち重要なコードストリームに対しては重点的に保護する。UEP は非重要なコードストリームのノイズに耐える性能を低下させるが、システムのビット誤りを防ぐ全体的な性能の向上には役立つ。

【0032】

40

2) デジタルファウンテンコード ( Digital Fountain Code ) は、伝送過程において、フィードバックおよび自動再送信のメカニズムを必要としないため、信号送受信による時間遅延およびブロードキャスト応用におけるフィードバック爆発の問題を回避している。デジタルファウンテンの基本的な思想は、以下のとおりである。送信側は、オリジナルデータを  $k$  個のデータシンボルに分割し、これらのデータシンボルについて符号化し、任意の長さの符号化シンボルのコードストリームを出力する。受信側は、 $n$  (  $n$  は  $k$  よりやや大きい ) 個の符号化シンボルを正確に受信するだけで全ての  $k$  個のデータシンボルを大きな確率で復元することができる。

【0033】

デジタルファウンテンコード自体は UEP 特性を有しており、重要性の異なるデータに

50

対する保護を実現することができる。従来の固定ビットレートのチャネル符号化方法に比べ、デジタルファウンテンコードは明らかに以下の優位性を有する。

【 0 0 3 4 】

a、理想的な拡張可能性を有する。単方向ブロードキャストはフィードバックがないため、送信側はユーザー数の増加によっていかなる影響も受けない。したがって、送信側は任意の数のユーザーにサービスを提供することができる。

【 0 0 3 5 】

b、経時変化のチャネルに適し、チャネルの容量を効率的に利用する。ユーザーの復号化特性は、チャネルの削除確率および帯域幅に関連しない。チャネルのパケットロス率が高く、状況が悪い場合でも、受信側の復号化に影響を与えず、即ち受信側は十分な数の符号化データさえ受信すれば正常に復号化することができ、より高い適応性を有する。

10

【 0 0 3 6 】

c、符号化と復号化の複雑度が低い。理想的な状況において、ファウンテンコードにより生成される各符号化シンボルは、線形の符号化と復号化の複雑度を有している。したがって、送受信側のコーデックの設計の簡略化およびソフトウェア化の実現に役立つ。

【 0 0 3 7 】

d、ヘテロジニアスなユーザーに対する適用性が高い。ファウンテンコードのビットレート特性は、異なるパケットロス率または帯域幅を有するユーザー同士が互いに影響せず、高品質のユーザーが低品質のユーザーの影響を受けないようにする。この他、デジタルファウンテンコードは、中断後の伝送再開、非同期アクセス等の複数のサービスモードをサポートする。

20

【 0 0 3 8 】

図 1 は、メディアリソースデータの F E C 二層構造である。図 1 に示すように、第一層では `source packet block` を多くの小さいブロックに分割し、それぞれについて F E C 保護を行い、第二層ではブロック全体に対して F E C 保護を行う。第一層における分割が細かいため、小さい遅延を提供することができ、第二層では復元性および小さい冗長を保証している。しかしながら、この方法では、F E C 符号化を 2 回行う必要があるため、符号化の効率が低下し、柔軟性が不十分である。

【 0 0 3 9 】

30

1 つのデータストリームは、その内容に応じて異なるタイプのデータパケットに分けることができる。異なるタイプのデータパケットは、異なる重要度を有する。例えば、データパケットにおける I、B、P フレームの重要度を図 2、3 に示す。1 つの画像群における各フレームの依存関係は、1 つの画像群における異なるフレームの依存程度および重要度が異なることを示し、I フレームは、最も重要であり、B、P フレームの重要度が比較的低い。そのため、データ内容の重要性に基づいて不均一誤り保護を行うことができる。具体的には、2 つの重要レベルを有するソースデータを伝送する状況を考慮する場合、ソースデータを前記標準に従って区画し、重要度に従って再配列し、システムファウンテンコードコーディング行列の構造を変更することにより、異なる内容に対する不均一誤り保護を実現する。そのシステムの基本的な構成を図 5 に示す。システムファウンテンコードの不均一誤り保護符号化により生成したデータを図 4 に示す。

40

【 0 0 4 0 】

[ 実施例 1 ]

本実施例では、R a p t o r Q システムコードを例とし、システム R a p t o r Q コードのコーディング行列の構造を図 6 に示す。ここで、D がソースデータ、C が中間コード、G \_ L D P C が L D P C 行列、I \_ S、I \_ H が単位行列、G \_ E N C が L T コーディング行列を表示する場合、コーディング行列は、以下のように表され得る。

【 数 4 4 】

50

$$G_p = \begin{bmatrix} G_{LDPC,1} & I_S & G_{LDPC,2} \\ G_{HDPC} & & I_H \end{bmatrix} \quad (1)$$

$$A = \begin{bmatrix} G_p \\ G_{ENC} \end{bmatrix} \quad (2)$$

【 0 0 4 1 】

F E C 符号化を行うときに、ソースデータおよび図 6 に示されるコーディング行列から中間コードを生成する。

【 数 4 5 】

$$C = A^{-1} \times \begin{bmatrix} 0 \\ D \end{bmatrix} \quad (3)$$

【 0 0 4 2 】

さらに、コーディング行列における L T 行列および得られた中間コードに基づいてソースデータおよび復元データを生成する。R は符号化して得られた復元データを示し、G \_ E N C ' は G \_ E N C と同じ生成方式により生成した L T 行列である。

【 数 4 6 】

$$\begin{bmatrix} D \\ R \end{bmatrix} = \begin{bmatrix} G_{ENC} \\ G_{ENC}' \end{bmatrix} \times C \quad (4)$$

【 0 0 4 3 】

上記方法によりソースデータを符号化する場合には、ソースデータに対して優先順位付けを行い、不均一誤り保護符号化を採用することができないため、コーディング行列 A および L T 行列 G \_ E N C の行列構造を改めて設計する必要がある。ソースデータに優先順位が異なる 2 つのデータがある場合を例とし、優先順位をそれぞれ優先順位 1、優先順位 2 に設定し、ソースデータにおける優先順位が異なるデータを区分し、それぞれに異なる冗長度を割り当てることにより、不均一誤り保護の効果が達成される。

【 0 0 4 4 】

< 符号化過程 >

そのコーディング行列構造を図 7 に示す。そのコーディング行列を一般式 ( 6 ) で表す。ここで、G \_ E N C <sub>1</sub> と G \_ E N C <sub>2</sub> は、異なるルールで生成したコーディング行列であり、重要なデータの保護を保証するために、G \_ E N C <sub>1</sub> 行列の度分布を高めることができる。

【 数 4 7 】

$$A_i = \begin{bmatrix} G_{p_i} \\ G_{ENC_i} \end{bmatrix} \quad i = 1, 2 \quad (5)$$

$$A = \begin{bmatrix} G_{p_1} & 0 \\ G_{ENC_1} & 0 \\ 0 & G_{p_2} \\ 0 & G_{ENC_2} \end{bmatrix} \quad (6)$$

このようにして、重要度が異なるデータに対して対応する中間コードを生成することができる。

【 数 4 8 】

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 \\ 0 & A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \end{bmatrix} \quad (7)$$

10

20

30

40

50

## 【 0 0 4 5 】

重要度が異なるデータに対する不均一誤り保護を達成するために、図 8 に示すように L T コーディング行列の構造を変更することで、重要なデータを保護するための冗長データを増加させることができる。ここで、 $G\_ENC_1'$  と  $G\_ENC_1$  は、生成方式が同じであり、 $G\_ENC_2'$  と  $G\_ENC_2$  は、生成方式が同じである。得られた復元データ  $R_1$  は、優先順位が 1 のデータにのみ関連し、復元データ  $R$  は、優先順位が 1 および 2 のデータに関連する。これによって、重要データの冗長度が高くなり、それに対する保護強度が高くなる。チャンネル条件の制限により、データの総冗長度が限られ、生成される  $R_1$  と  $R_2$  の比は、 $[G\_ENC_1' \ 0]$  と  $[G\_ENC_1' \ G\_ENC_2']$  の行数比により確定されるので、 $[G\_ENC_1' \ 0]$  の行数を増加することにより優先順位が 1 のデータに対する保護強度を向上させることができる。

10

## 【 数 4 9 】

$$\begin{bmatrix} D_1 \\ D_2 \\ R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} G\_ENC_1 & 0 \\ 0 & G\_ENC_2 \\ G\_ENC_1' & 0 \\ G\_ENC_1'' & G\_ENC_2' \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (8)$$

## 【 0 0 4 6 】

20

< 復号化過程 >

損失チャンネル ( Lossy channel ) を経過した後、一部のデータパケットがロスするので、受信したシンボル  $ISI$  に基づいて復元行列を生成するために、ロスシンボル  $ISI$  に対応する行列の行を消去する必要がある。例えば、行列  $REV\_G\_ENC_1$  の場合、受信した  $REV\_D_1$  および送信端の  $D_1$  に基づいて、ロスデータの  $ISI$  を得ることができ、これによって、 $G\_ENC_1$  に対応する行を削除することができる。符号化過程と同様に、第 1 段階の復号化の際に、復元行列および受信したデータパケットに基づいて中間コードワードを得る。

## 【 数 5 0 】

30

$$\begin{bmatrix} G_{p1} & 0 \\ REV\_G\_ENC_1 & 0 \\ 0 & G_{p2} \\ 0 & REV\_G\_ENC_2 \\ REV\_G\_ENC_1' & 0 \\ \_REV\_G\_ENC_1'' & REV\_G\_ENC_2' \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ 0 \\ REV\_D_2 \\ REV\_R_1 \\ REV\_R_2 \end{bmatrix} \quad (9)$$

## 【 0 0 4 7 】

上記行列方程式を一次行変換して以下の形式を得る。

## 【 数 5 1 】

40

$$\begin{bmatrix} G_{p1} & 0 \\ REV\_G\_ENC_1 & 0 \\ REV\_G\_ENC_1' & 0 \\ 0 & G_{p2} \\ 0 & REV\_G\_ENC_2 \\ \_REV\_G\_ENC_1'' & REV\_G\_ENC_2' \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \end{bmatrix} \quad (10)$$

ここで、

## 【 数 5 2 】

50

$$REV\_A = \begin{bmatrix} G\_p_1 & 0 \\ REV\_G\_ENC_1 & 0 \\ REV\_G\_ENC_1' & 0 \\ 0 & G\_p_2 \\ 0 & REV\_G\_ENC_2 \\ \_REV\_G\_ENC_1'' & REV\_G\_ENC_2' \end{bmatrix}$$

にし、受信した各優先順位でのデータパケットの数および受信したデータパケットの総数に基づいてケースバイケースで復号化する。

10

【 0 0 4 8 】

受信した符号化シンボルの総数はソースデータ符号化シンボルの数よりも少なく、かつ、  
【 数 5 3 】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_1' \end{bmatrix}$$

行数は列数よりも少なく、 $C_1$ 、 $C_2$  はいずれも算出できない。

【 0 0 4 9 】

20

受信した符号化シンボルの総数はソースデータ符号化シンボルの数よりも少ないが、  
【 数 5 4 】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_1' \end{bmatrix}$$

の行数は列数以上である。

【 数 5 5 】

30

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_1' \end{bmatrix}$$

の行数が列数以上である場合、 $REV\_A_1$  はリバーシブルであり、その逆行列は  $REV\_A_1^{-1}$  と記し、

【 数 5 6 】

$$[C_1] = REV\_A^{-1} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \end{bmatrix} \quad (11)$$

40

は算出されるが、 $C_2$  は算出できない。

【 0 0 5 0 】

受信した符号化シンボルの総数はソースデータ符号化シンボルの数よりも大きく、かつ  
【 数 5 7 】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_1' \end{bmatrix}$$

50



の行数は列数以上である。

【数 5 8】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_1' \end{bmatrix}$$

の行数が列数以上である場合、 $REV\_A_1$  はリバーシブルであり、その逆行列は  $REV\_A_1^{-1}$  と記す。

【数 5 9】

10

$$REV\_A_2 = \begin{bmatrix} G\_p_2 \\ REV\_G\_ENC_2 \\ REV\_G\_ENC_2' \end{bmatrix}$$

の行数が列数以上である場合、 $REV\_A_2$  はリバーシブルであり、その逆行列は  $REV\_A_2^{-1}$  と記し、

【数 6 0】

$$B_{1,1} = \begin{bmatrix} 0 \\ 0 \\ REV\_G\_ENC_1'' \end{bmatrix}$$

20

であり、

【数 6 1】

$$REV\_A^{-1} = \begin{bmatrix} REV\_A_1^{-1} & 0 \\ REV\_A_2^{-1} B_{1,1} REV\_A_1^{-1} & REV\_A_2^{-1} \end{bmatrix}$$

30

であることが分かる。これによって、

【数 6 2】

$$\begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = REV\_A^{-1} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \end{bmatrix} = \begin{bmatrix} REV\_A_1^{-1} & 0 \\ REV\_A_2^{-1} B_{1,1} REV\_A_1^{-1} & REV\_A_2^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \end{bmatrix} \quad (12)$$

となる。

しかし、

【数 6 3】

40

$$REV\_A_2 = \begin{bmatrix} G\_p_2 \\ REV\_G\_ENC_2 \\ REV\_G\_ENC_2' \end{bmatrix}$$

の行数が列数よりも小さい場合、 $REV\_A_2$  には逆行列がなく、

【数 6 4】

50

$$[C_1] = REV\_A^{-1} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \end{bmatrix}$$

しか算出できない。

【 0 0 5 1 】

受信した符号化シンボルの総数がソースデータ符号化シンボルの数以上である場合、

【数 6 5 】

$$REV\_A_1 = \begin{bmatrix} G\_p_1 \\ REV\_G\_ENC_1 \\ REV\_G\_ENC_1' \end{bmatrix}$$

10

の行数が列数よりも小さいが、

【数 6 6 】

$$REV\_A_2 = \begin{bmatrix} G\_p_2 \\ REV\_G\_ENC_2 \\ REV\_G\_ENC_2' \end{bmatrix}$$

20

の行数が列数よりも大きい。

ガウスの消去法により行列  $REV\_A$  を直接解き、即ち、以下の線型方程式系を解くことにより、 $C_1$ 、 $C_2$  が求められる。

【数 6 7 】

$$\begin{bmatrix} G\_p_1 & 0 \\ REV\_G\_ENC_1 & 0 \\ REV\_G\_ENC_1' & 0 \\ 0 & G\_p_2 \\ 0 & REV\_G\_ENC_2 \\ \_REV\_G\_ENC_1'' & REV\_G\_ENC_2' \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \end{bmatrix} \quad (13)$$

30

【 0 0 5 2 】

上記デコード過程の第 1 段階から分かるように、如何なる場合においてもまず  $C_1$  を正確に算出する必要がある。中間コードワード  $C_1$  および  $C_2$  は、デコードの第 2 段階でソースデータの重要なデータを復元するので、 $C_1$  のデコード成功率を高めることにより、ソースデータの正確な復元の確率が高くなる。

【 0 0 5 3 】

得られた中間コードワードに基づいてシステムコード符号化シンボルを生成する。

【数 6 8 】

$$\begin{bmatrix} D_1 \\ D_2 \end{bmatrix} = \begin{bmatrix} G\_ENC_1 & 0 \\ 0 & G\_ENC_2 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (14)$$

40

第 1 段階で  $C_1$ 、 $C_2$  が算出できない場合、復号化が失敗し、ソースシンボルが復号化されない。 $C_1$  のみが算出できる場合、優先順位が高いソースデータ  $D_1$  が正確に復号化されることを保証することができる。 $C_1$ 、 $C_2$  がともに算出される場合、上式により最終的なソースコード  $D_1$  および  $D_2$  を得ることができる。

【 0 0 5 4 】

また、この実施形態は、2 つの優先順位のシーンにのみ適用するだけでなく、複数の

50

優先順位のソースデータにも適用される。具体的な過程は、以下の通りである。

【 0 0 5 5 】

< 符号化過程 >

まず、ソースデータおよびソースデータの優先順位に基づいて中間コードワードを生成する。ソースデータが 1 個のデータパケットに分けられる場合、

【 数 6 9 】

$$\begin{bmatrix} G_{p_1} & 0 & \dots & 0 \\ G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{p_2} & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{p_l} \\ 0 & 0 & \dots & G_{ENC_l} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_l \end{bmatrix} \quad (15) \quad 10$$

が得られる。式 ( 1 6 ) により中間コードワード  $C_1, C_2, \dots, C_l$  が得られる。

【 数 7 0 】

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_l^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_l \end{bmatrix} \quad (16) \quad 20$$

得られた中間コードワードに基づいてソースデータおよび復元データを生成する。

【 数 7 1 】

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_l \\ R_1 \\ R_2 \\ \dots \\ R_l \end{bmatrix} = \begin{bmatrix} G_{ENC_1} & 0 & \dots & 0 \\ 0 & G_{ENC_2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G_{ENC_l} \\ G_{ENC_{11}} & 0 & \dots & 0 \\ G_{ENC_{12}} & G_{ENC_{21}} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC_{1l}} & G_{ENC_{2(l-1)}} & \dots & G_{ENC_{l1}} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_l \end{bmatrix} \quad (17) \quad 30$$

【 0 0 5 6 】

符号化過程全体を図 9 に示す。ソースデータおよび復元データを送信端より受信端に送信し、チャンネル減衰のため、一部のデータがロスする可能性があり、受信端がデータを受信した後、受信したデータの状況に基づいて復号化してソースデータを復元する。

【 0 0 5 7 】

< 復号化過程 >

一部のデータがロスしたので、受信したシンボル  $ISI$  に基づいて復元行列を生成する必要がある。つまり、ロスしたシンボル  $ISI$  に対応する行列行を消去し、対応する行列 (  $REV\_G\_ENC$  と記す ) を生成する。受信したデータの数に応じて、ケースバイケースで中間コードワードを生成する。

【 0 0 5 8 】

【 数 7 2 】

40

$$\begin{bmatrix}
 G_{p_1} & 0 & \dots & 0 \\
 REV\_G\_ENC_1 & 0 & \dots & 0 \\
 0 & G_{p_2} & \dots & 0 \\
 0 & REV\_G\_ENC_2 & \dots & 0 \\
 \dots & \dots & \dots & \dots \\
 0 & 0 & \dots & G_{p_l} \\
 0 & 0 & \dots & REV\_G\_ENC_l \\
 REV\_G\_ENC_{11} & 0 & \dots & 0 \\
 REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\
 \dots & \dots & \dots & \dots \\
 REV\_G\_ENC_{1l} & REV\_G\_ENC_{2(l-1)} & \dots & REV\_G\_ENC_{l1}
 \end{bmatrix}
 \begin{bmatrix}
 C_1 \\
 C_2 \\
 \dots \\
 C_l
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 REV\_D_1 \\
 0 \\
 REV\_D_2 \\
 \dots \\
 0 \\
 REV\_D_l \\
 REV\_R_1 \\
 REV\_R_2 \\
 \dots \\
 REV\_R_l
 \end{bmatrix}
 \quad (18)$$

10

上の行列方程式を一次行変換して以下の形式が得られる。

【数 7 3】

$$\begin{bmatrix}
 G_{p_1} & 0 & \dots & 0 \\
 REV\_G\_ENC_1 & 0 & \dots & 0 \\
 REV\_G\_ENC_{11} & 0 & \dots & 0 \\
 0 & G_{p_2} & \dots & 0 \\
 0 & REV\_G\_ENC_2 & \dots & 0 \\
 REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\
 \dots & \dots & \dots & \dots \\
 0 & 0 & \dots & G_{p_l} \\
 0 & 0 & \dots & REV\_G\_ENC_l \\
 REV\_G\_ENC_{1l} & REV\_G\_ENC_{2(l-1)} & \dots & REV\_G\_ENC_{l1}
 \end{bmatrix}
 \begin{bmatrix}
 C_1 \\
 C_2 \\
 \dots \\
 C_l
 \end{bmatrix}
 =
 \begin{bmatrix}
 0 \\
 REV\_D_1 \\
 REV\_R_1 \\
 0 \\
 REV\_D_2 \\
 REV\_R_2 \\
 \dots \\
 0 \\
 REV\_D_l \\
 REV\_R_l
 \end{bmatrix}
 \quad (19)$$

20

【数 7 4】

$$REV\_A = \begin{bmatrix}
 G_{p_1} & 0 & \dots & 0 \\
 REV\_G\_ENC_1 & 0 & \dots & 0 \\
 REV\_G\_ENC_{11} & 0 & \dots & 0 \\
 0 & G_{p_2} & \dots & 0 \\
 0 & REV\_G\_ENC_2 & \dots & 0 \\
 REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\
 \dots & \dots & \dots & \dots \\
 0 & 0 & \dots & G_{p_l} \\
 0 & 0 & \dots & REV\_G\_ENC_l \\
 REV\_G\_ENC_{1l} & REV\_G\_ENC_{2(l-1)} & \dots & REV\_G\_ENC_{l1}
 \end{bmatrix}$$

30

とし、 $R(i) (1 \leq i \leq l)$  を受信して拡張した後の第  $i$  レベルのシンボル数、 $L(i)$  を符号化端の各レベルのシンボルの長さとし、 $i = 1 : l$  に対して、各レベルで生成した行列のランク  $eachRank(i) = \min(Len(R(i)), sum(L(1:i)))$  を計算し、さらに、累積行列のランク  $cumRank(i) = \min(cumRank(i-1) + eachRank(i), sum(L(1:i)))$  を計算する。ここで、 $cumRank(1) = eachRank(1)$  である。順に  $cumRank(i) \geq sum(L(1:i))$  を判断し、成立した場合、 $cumFullRank(i) = 1$  であり、成立しない場合、 $cumFullRank(i) = 0$  である。以下、受信した各レベルのシンボル数に応じて、ケースバイケースで中間コード  $C_1, C_2 \dots C_l$  を求める。ここで、 $i$  初期値を 1 に設定する。

40

$cumFullRank(i) = 1 \ \&\& \ cumFullRank(1:i) > 0$  を満たした場合、RFC 6330 復号化方法により  $C(1) \sim C(i)$  を解く。

【数 7 5】

50

$$REV\_A_{\Delta i} = \begin{bmatrix} G\_p_1 & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ 0 & G\_p_2 & \dots & 0 \\ 0 & REV\_G\_ENC_2 & \dots & \dots \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G\_p_i \\ 0 & 0 & \dots & REV\_G\_ENC_i \\ REV\_G\_ENC_{1i} & REV\_G\_ENC_{2(i-1)} & \dots & REV\_G\_ENC_{i1} \end{bmatrix} \quad 10$$

とすると、

【数 7 6】

$$REV\_A_{\Delta i}^{-1} = \begin{bmatrix} A_1^{-1} & 0 & 0 & 0 \\ X_{1,2} & A_2^{-1} & 0 & 0 \\ \dots & \dots & \dots & 0 \\ X_{1,i} & X_{2,(i-1)} & \dots & A_{i-1}^{-1} \end{bmatrix}$$

である。ここで、

【数 7 7】

$$X_{1,2} = A_2^{-1} B_{1,2} A_1^{-1}, \quad X_{a,b} = A_b^{-1} B_{a,b} A_a^{-1} + \sum_{k=a+1}^{b-1} A_b^{-1} B_{k,b} X_{a,k}$$

である。さらに、中間コードは、下式により得られる。

【数 7 8】

$$C_{i\_temp} = REV\_A_i^{-1} \begin{bmatrix} 0 \\ REV\_D_i \\ REV\_R_i \end{bmatrix},$$

$$C_1 = C_{1\_temp}$$

$$C_2 = C_{2\_temp} + X_{1,2} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \end{bmatrix} = C_{2\_temp} + A_2^{-1} B_{1,2} C_1$$

$$C_3 = C_{3\_temp} + X_{1,3} \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \end{bmatrix} + X_{2,3} \begin{bmatrix} 0 \\ REV\_D_2 \\ REV\_R_2 \end{bmatrix} = C_{2\_temp} + A_3^{-1} B_{1,3} C_1 + A_3^{-1} B_{2,3} C_2$$

さらに、数学的帰納法により

【数 7 9】

$$C_i = C_{i\_temp} + \sum_{j=1}^{i-1} X_{j,i} \begin{bmatrix} 0 \\ REV\_D_j \\ REV\_R_j \end{bmatrix} = C_{i\_temp} + \sum_{j=1}^{i-1} A_i^{-1} B_{j,i} C_j$$

$C_1, C_2 \dots C_i$  が算出できるが、 $C_{i+1} \dots C_L$  が算出できない。

$cumFullRank(i) = 1$  を満足し、 $cumFullRank(1:i) > 0$  を満足しない場合、ガウス消去復号化法により  $C(1) \sim C(i)$  を解く。

【数 8 0】

$$REV\_A_{\Delta i} = \begin{bmatrix} G\_p_1 & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ 0 & G\_p_2 & \dots & 0 \\ 0 & REV\_G\_ENC_2 \dots & \dots & \dots \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G\_p_i \\ 0 & 0 & \dots & REV\_G\_ENC_i \\ REV\_G\_ENC_{1i} & REV\_G\_ENC_{2(i-1)} & \dots & REV\_G\_ENC_{i1} \end{bmatrix} \quad 10$$

とすると、ガウスの消去法により線型方程式系：

【数 8 1】

$$\begin{bmatrix} G\_p_1 & 0 & \dots & 0 \\ REV\_G\_ENC_1 & 0 & \dots & 0 \\ REV\_G\_ENC_{11} & 0 & \dots & 0 \\ 0 & G\_p_2 & \dots & 0 \\ 0 & REV\_G\_ENC_2 \dots & \dots & \dots \\ REV\_G\_ENC_{12} & REV\_G\_ENC_{21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & G\_p_i \\ 0 & 0 & \dots & REV\_G\_ENC_i \\ REV\_G\_ENC_{1i} & REV\_G\_ENC_{2(i-1)} & \dots & REV\_G\_ENC_{i1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_i \end{bmatrix} = \begin{bmatrix} 0 \\ REV\_D_1 \\ REV\_R_1 \\ 0 \\ REV\_D_2 \\ REV\_R_2 \\ \dots \\ 0 \\ REV\_D_i \\ REV\_i \end{bmatrix} \quad (20) \quad 20$$

を解くことにより、 $C_1$ 、 $C_2 \dots C_i$ を算出できるが、 $C_{i+1} \dots C_L$ を算出できない。

【0059】

ある  $i$  値の場合、 $cumFullRand(i) == 1$  を満足しない場合、対応する  $C_i$  を算出できない。 $i == 1$  まで前記 2 つの条件のいずれも満足できない場合、 $C_1$ 、 $C_2 \dots C_L$  はいずれも算出できず、復号化が失敗する。

【0060】

$i = i - 1$  を引き続き循環させ、即ち  $i = L : -1 : 1$  にすることにより、上記過程を繰り返す。

【0061】

復号化プロセス全体を図 10 に示す。

本実施例の方法によれば、資源が有効に節約される。受信端で、場合によって受信したデータ（例えば、B フレーム）を人為的に捨てると、伝送資源の浪費をもたらす。上記実施形態によれば、ソース端から問題が解決され、望ましくないパケットを伝送過程でより大きな確率でロスさせることにより、重要なパケットがより大きな程度で保護される。

【0062】

本実施例の方法によれば、個性的な伝送方式が提供される。チャンネル状況、ユーザ体験などに基づいて不均一誤り保護の態様を設計することにより、ビデオ伝送がより柔軟で緻密になる。また、送信端の符号化方式が柔軟になっても、受信端は、リスト情報に基づいてオリジナルデータを復元することができる。

【0063】

本実施例の方法によれば、より柔軟なコーディング行列の設計が提供される。実際のメディアの応用シーンに応じて、メディア内容に基づいてデータの優先順位付けを行い、実際の必要に応じて FEC コーディング行列をより柔軟に設計することができ、これによって、不均一誤り保護の効果が達成される。

【0064】

本実施例の方法によれば、より柔軟な復号化方式が提供される。受信した各優先順位の

30

40

50

データの数に応じて、異なる復号化アルゴリズムを採用することにより、限られた帯域幅資源下で、優先順位が高いデータが最大限に復元され、優先順位が比較的高いデータに対する保護強度が高くなる。

【 0 0 6 5 】

[ 実施例 2 ]

ソースデータパケットにおける各サブパケットをそれぞれ異なる所定優先順位にマッピングし、または実施例 1 と実施例 2 とを組み合わせることができる。具体的には、以下の通りである。

【 0 0 6 6 】

図 1 1 は、メディア内容に基づく自己適応システムコード F E C システムのフレーム図である。前記システムは、サーバーおよび少なくとも 1 つの端末を含む。まず、サーバー側は、実際の応用シーンおよびメディア内容に基づいて複数の優先順位を分け、データパケットのヘッダにおける前記データパケットに関連する重要な情報を解析し、各データパケットを異なる優先順位にマッピングする。さらに、優先順位状況およびマッピング関係に基づいて、前方誤り訂正符号化 ( F E C ) を必要とするデータパケットを優先順位に従って高から低に再配列する。再配列されたソースデータに対して、実際のチャンネル状況に応じて、不均一誤り保護の F E C 符号化方式により符号化を行い、復元データ ( r e p a i r s y m b o l s ) を生成する。F E C 符号化終了後に、ソースデータパケットおよび復元データパケットに現在のデータパケットシーケンスを指示するインデックスフィールドを加える。各優先順位でのソースデータおよび復元データを統計し、対応するシグナリング情報を生成し、シグナリングにおいて各優先順位でのデータパケットの数または割合、および各優先順位で対応する復元データの数または割合を指示することにより、受信端がデータを受信した後にデータを正確に復元することができる。最後に、ソースデータパケット、復元データパケットおよびシグナリングを端末 ( クライアントまたは受信端 ) に送信する。

【 0 0 6 7 】

対応するように、端末側は、まずシグナリング情報および受信したデータパケットにおけるインデックスフィールドを解析し、各優先順位で受信したソースデータパケットおよび復元データパケットの数および対応するデータパケットインデックを得る。さらに、得られたデータパケットインデック情報に基づいて受信端でコーディング行列を生成する。最後に、生成したコーディング行列に基づいてケースバイケースで F E C 復号化を行い、ソースデータパケットを復元する。復元したソースデータパケットのシーケンスが再配列されたシーケンスであるので、データパケットのヘッダにおけるソースデータパケットのシーケンスを解析し、ソースデータパケットを再配列することにより、ソースデータパケットの初期データが得られる。

【 0 0 6 8 】

以下、具体的な実施例により本発明の実施形態を詳しく説明する。以下の具体的な実施例は互いに組み合わせることができ、同じまたは類似の概念または過程は、いくつかの実施例は省略される。

【 0 0 6 9 】

( 実施例 2 . 1 )

図 1 2 に示すように、以下のステップを含むことができる。

S 1 : 符号化してパッケージされたデータパケットのヘッダを解析し、D U h e a d e r における p r i o r i t y フィールドを読み取り、実際の応用に応じて実際に採用される優先順位の数および各優先順位での p r i o r i t y フィールドとのマッピング関係を確定する。

【 0 0 7 0 】

S 2 : 分けられた優先順位およびマッピング関係に基づいて、オリジナルのソースデータストリームを優先順位に従って高から低に再配列し、各優先順位でのデータパケットの数を統計する ( 図 1 3 ) 。

## 【 0 0 7 1 】

S 3 : 再配列されたソースデータストリームに対して、システム R a p t o r Q システムコードの自己適応 F E C 符号化スキームを採用して F E C 符号化を行い、復元データを生成する。

## 【 0 0 7 2 】

具体的には、前記中間データパケットに対して、R a p t o r Q システムコードの自己適応 F E C 符号化方式を採用して F E C 符号化を行い、以下の中間コードワードを得る。

## 【 数 8 2 】

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

10

ここで、 $A_i$  は、第  $i$  のソースデータパケットに対応するコーディング行列を示す。 $C_1, C_2, \dots, C_i, \dots, C_N$  は、それぞれ  $N$  種類のソースデータパケットの中間コードに対応する。 $D_1, D_2, \dots, D_i, \dots, D_N$  は、それぞれ  $N$  種類のソースデータパケットのデータに対応する。上付き  $-1$  は、逆行列の演算子である。

20

前記中間コードワードに基づいて復元データパケットを取得し、前記復元データパケットに F E C p a y l o a d I D を追加する。前記復元データパケットにおける復元データは、以下の通りである。

## 【 数 8 3 】

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

30

ここで、 $R_1, R_2, \dots, R_i, \dots, R_N$  は、それぞれ  $1 \sim N$  個のソースデータパケットの復元データに対応する。 $G_{ENCi}$  は、第  $i$  のソースデータパケットに対応する L T コーディング行列を示す。 $G_{ENC1i}$  は、 $G_{ENC1}$  と同じ生成ルールを有する L T コーディング行列を示す。 $G_{ENCi1}$  は、 $G_{ENCi}$  と同じ生成ルールを有する L T コーディング行列を示す。ここで、 $i = 1, 2, \dots, N$  である。

## 【 0 0 7 3 】

40

F E C エンコーダーから出力されたデータストリーム（再配列された後のデータストリーム）に基づいてソースデータパケットおよび復元データパケットを生成し、F E C p a y l o a d I D を追加する（図 1 4）。

## 【 0 0 7 4 】

S 4 : A L - F E C m e s s a g e を生成し、シグナリングにおいて各優先順位でのデータパケットの数および各優先順位での対応する復元データパケットの数を指し示すことにより、受信端がデータを受信した後にデータを正確に復元することができる。具体的なシグナリングを表 1 に示す。

## 【 0 0 7 5 】

50



【表 1】

シグナリング情報照合

[illegible]

10

【 0 0 7 6 】

表1において、`number_of_priority`は優先順位の数を示す。`fec_priority_value`(8ビット)は、各優先順位に対応する値を示し、この値が多くともDU headerにおける`priority`で定義された優先順位と同じであってもよく、複数のDU headerにおける`priority`に対応してもよい。`number_of_source_symbol`(16ビット)は、優先順位が`fec_priority_value`で示される値でのソースシンボルの数を示す。`number_of_repair_symbol`(16位)は、優先順位が`fec_priority_value`で示される値での復元シンボルの数を示す。`private_fec_flag`は、指示ビットであり、プライベートFEC符号化スキームを使用するか否かを明示する。`private_flag`は、指示ビットであり、使用されるプライベートFEC符号化スキームを記述する1つの`private_field`が存在するか否かを明示する。`private_field_length`は、プライベートFEC符号化スキームのフィールドの長さを記述する長さフィールドである。`private_field`は、プライベートFECスキームの詳細情報を記述する。`priority_id`は、MMTパケットの優先順位を指示する優先順位idである。`fec_code_id_for_repair_flow`は、使用されるFEC符号化スキームを記述する。`repair_flow_id`(8ビットの整数)は、生成したFEC repair flowとFEC repairパケットのヘッダにおける`packet_id`との間に対応関係を有することを示す。`maximum_k_for_repair_flow`(24ビットの整数)は、`source_symbol`ブロックにおける`source_symbol`の最大数を示す。`maximum_p_for_repair_flow`(24ビットの整数)は、`repair_symbol`ブロックにおける`repair_symbol`の最大数を示す。`protection_window_time`は、FEC符号化において最初の`source`または`repair`パケットの送信と、最後の`source`または`repair`パケットの送信との間の最大時間差を示す保護ウィンドウ時間(単位:ミリ秒)である。`protection_window_size`は、FEC符号化ストリームにおいて最初のFECパケットを送信する負荷と最後のFECパケットを送信する負荷との間の最大カウント値を示す保護ウィンドウ値である。

20

30

40

【 0 0 7 7 】

図 15 に示すように、FEC 各優先順位でのデータパケットの数に基づいて、下式によりソースデータに対応するコーディング行列の対応行のインデックを得る。

index = FEC payload ID - number of packets of

50

all priorities before this packet.

【0078】

式中、indexは、ソースデータのコーディング行列の対応行のインデックを示す。FEC payload IDは、データパケットのヘッダのインデック情報を示す。number of packets of all priorities before this packetは、前記データパケットの前のすべての優先順位のデータパケットの総数を示す。

【0079】

ここで、各優先順位の数が100である4つの優先順位を例として説明する(図16)。source\_FECpayload\_IDは、再配列された後のソースデータパケットのシーケンスに従って配列したものである。優先順位が1である場合、現在のsource\_FECpayload\_IDに基づいてコーディング行列を生成するISI(インデック)を直接確定することができる。優先順位が1よりも大きい場合、ヘッダを解析する際に、そのsource\_FECpayload\_IDの値および各優先順位でのソースシンボルの数に基づいてどの優先順位に属するかを確定することができる。例えば、source\_FECpayload\_ID=108である場合、優先順位2に属することを確定することができ、上記一般式によりそのISI=8であることが分かる。復元データパケットについては、ヘッダにおけるrepair\_FECpayload\_IDに基づいて生成行列のISIを確定することができる。送信端と受信端は同じFEC符号化メカニズムを採用するので、前記2つのインデックに基づいて対応するコーディング行列を生成することができる。

【0080】

生成したFECコーディング行列に基づいて、コーディング行列を一次行変換し、対応して受信したデータパケットを配列し、repair\_FECpayload\_IDの値および各優先順位での復元シンボルの数に基づいてどの優先順位に属するかを確定することにより、シンボルブロックにおける復元シンボルの位置を調整し、同じ変化を達成する(図17)。受信した異なる優先順位のデータパケットの数によって、ケースバイケースでデコードを行い、そのフローチャートを図10に示す。最終的に復元されたソースデータを取得し、さらにヘッダにおけるpacket\_sequence\_numberに基づいて得られたソースデータパケットを初期シーケンスに復元する(図18)。

【0081】

(実施例2.2)

図12に示すように、以下のステップを含むことができる。

S1: 符号化してパッケージされたデータパケットのヘッダを解析し、DU headerにおけるpriorityフィールドを読み取り、実際の応用に応じて実際に採用される優先順位の数および各優先順位でのpriorityフィールドとのマッピング関係を確定する。

【0082】

S2: 分けられた優先順位およびマッピング関係に基づいて、オリジナルのソースデータストリームを優先順位に従って高から低に再配列し、各優先順位でのデータパケットの数を統計する(図13)。

【0083】

S3: 再配列されたソースデータストリームに対して、システムRaptorQシステムコードの自己適応FEC符号化スキームを採用してFEC符号化を行い、復元データを生成する。

【0084】

具体的には、前記中間データパケットに対して、RaptorQシステムコードの自己適応FEC符号化方式を採用してFEC符号化を行い、以下の中間コードワードを得る。

【数84】

10

20

30

40

50

$$\begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix} = \begin{bmatrix} A_1^{-1} & 0 & \dots & 0 \\ 0 & A_2^{-1} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & A_N^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ D_1 \\ 0 \\ D_2 \\ \dots \\ 0 \\ D_N \end{bmatrix}$$

ここで、 $A_i$  は、第  $i$  のソースデータパケットに対応するコーディング行列を示す。 $C_1, C_2, \dots, C_i, \dots, C_N$  は、それぞれ  $N$  種類のソースデータパケットの中間コードに対応する。 $D_1, D_2, \dots, D_i, \dots, D_N$  は、それぞれ  $N$  種類のソースデータパケットのデータに対応する。上付き  $-1$  は、逆行列の演算子である。

前記中間コードワードに基づいて復元データパケットを取得し、前記復元データパケットに `FEC payload ID` を追加する。前記復元データパケットにおける復元データは、以下の通りである。

【数 8 5】

$$\begin{bmatrix} D_1 \\ D_2 \\ \dots \\ D_N \\ R_1 \\ R_2 \\ \dots \\ R_N \end{bmatrix} = \begin{bmatrix} G_{ENC1} & 0 & \dots & 0 \\ 0 & G_{ENC2} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & G_{ENCN} \\ G_{ENC11} & 0 & \dots & 0 \\ G_{ENC12} & G_{ENC21} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ G_{ENC1N} & G_{ENC2(N-1)} & \dots & G_{ENCN1} \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \\ \dots \\ C_N \end{bmatrix}$$

ここで、 $R_1, R_2, \dots, R_i, \dots, R_N$  は、それぞれ  $1 \sim N$  個のソースデータパケットの復元データに対応する。 $G_{ENCi}$  は、第  $i$  のソースデータパケットに対応する LT コーディング行列を示す。 $G_{ENC1i}$  は、 $G_{ENC1}$  と同じ生成ルールを有する LT コーディング行列を示す。 $G_{ENCi1}$  は、 $G_{ENCi}$  と同じ生成ルールを有する LT コーディング行列を示す。ここで、 $i = 1, 2, \dots, N$  である。

【0085】

S4：データパケットのヘッダ拡張において対応する指示情報を生成し、各優先順位でのデータパケットの数および各優先順位での対応する復元データパケットの数を指し示すことにより、受信端がデータを受信した後にデータを正確に復元することができる。詳細を表 2 に示す。

【0086】

【表 2】

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1			
+	+	+	+
	type		length
+	+	+	+
	N		SSBL[1]
+	+	+	+
	SSBL[2]		.....
+	+	+	+
	SSBL[N]		RSBL[1]
+	+	+	+
	RSBL[1]		RSBL[2]
+	+	+	+
	.....		RSBL[N]
+	+	+	+

【0087】

表 2 中、`type` (16 ビット) は、現在のヘッダ拡張のタイプを示し、ここで、自己適応前方誤り訂正符号化に用いられる中間データパケットまたは復元データパケットを示す。`length` (16 ビット) は、`header_extension` の長さを示す。`FEC priority` (8 ビット) は、現在のデータパケットが属する優先順位を指示す

10

20

30

40

50

る。sub\_encoded\_symbol\_id (16ビット) は、この優先順位での中間データパケットまたは復元データパケットの数のインデックを示す。N (8ビット) は、優先順位の総数を示す。SSBL[i] (24ビット) は、各優先順位のソースデータパケットの数を示し、ここで、i は第 i の優先順位を指示する。RSBL[i] (24ビット) は、i 表示第 i の優先順位、各優先順位の復元データパケットの数を示し、ここで、i は第 i の優先順位を指示する。

#### 【0088】

図19に示すように、FEC各優先順位でのデータパケットの数に基づいて、下式によりソースデータに対応するコーディング行列の対応行のインデックを得る。

index = FEC payload ID - number of packets of all priorities before this packet. 10

#### 【0089】

ここで、index は、ソースデータのコーディング行列の対応行のインデックを示す。FEC payload ID は、データパケットのヘッダのインデック情報を示す。number of packets of all priorities before this packet は、前記データパケットの前のすべての優先順位のデータパケットの総数を示す。

#### 【0090】

生成したFECコーディング行列に基づいて、コーディング行列を一次行変換し、対応して受信したデータパケットを配列し、repair\_FECpayload\_IDの値および各優先順位での復元シンボルの数に基づいてどの優先順位に属するかを確定することにより、シンボルブロックにおける復元シンボルの位置を調整し、同じ変化を達成する(図17)。受信した異なる優先順位のデータパケットの数によって、ケースバイケースでデコードを行い、そのフローチャートを図10に示す。最終的に復元されたソースデータを取得し、さらにヘッダにおけるpacket\_sequence\_numberに基づいて得られたソースデータパケットを初期シーケンスに復元する(図18)。

20

#### 【0091】

本実施例において、サーバー側は、データストリームにおける異なるデータを重要度に応じて分割することにより、重要度が異なるデータを異なる保護を与えることができる。送信したデータパケットのヘッダ情報およびシグナリング情報により標識する。システムコードの方式を採用し、チャンネル状況およびデータの重要度に応じてシステムファウンテンコードのコーディング行列を自己適応的に調整することにより、重要度が異なるデータに対する不均一誤り保護を達成する。端末側でシグナリング情報およびヘッダ情報を解析し、受信した異なる優先順位のデータの数に応じて柔軟な復号化アルゴリズムを採用することで柔軟な符号化および復号化を達成し、ヘッダ情報に基づいてソースデータストリームを最終的に復元する。従って、以下の有益な効果が得られる。

30

#### 【0092】

1) 資源が節約される。例えば、端末側(受信端)において場合によって受信したデータ(例えば、Bフレーム)を人為的に捨てると、伝送資源の浪費をもたらす。本実施例の方法によれば、ソース端から問題が解決され、望ましくないパケットを伝送過程でより大きな確率でロスさせることにより、重要なデータパケットがより大きな程度で保護される。

40

#### 【0093】

2) 個性的な伝送方式が達成される。例えば、チャンネル状況、ユーザ体験などに基づいて不均一誤り保護の態様を設計することにより、ビデオ伝送がより柔軟で緻密になる。また、送信端の符号化方式が柔軟になっても、受信端は、リスト情報に基づいてオリジナルデータを復元することができる。

#### 【0094】

3) より柔軟なコーディング行列の設計が達成される。例えば、実際のメディアの応用シーンに応じて、メディア内容に基づいてデータの優先順位付けを行い、実際の必要に応じてFECコーディング行列をより柔軟に設計することができ、これによって、不均一誤

50

り保護の効果が達成される。

【 0 0 9 5 】

4) より柔軟な復号化方式が達成される。例えば、受信した各優先順位のデータの数に応じて、異なる復号化アルゴリズムを採用することにより、限られた帯域幅資源下で、優先順位が高いデータが最大限に復元され、優先順位が比較的高いデータに対する保護強度が高くなる。

【 0 0 9 6 】

なお、本発明で提供されるメディア内容に基づく自己適応システムコード F E C 符号化および復号化方法のステップは、メディア内容に基づく自己適応システムコード F E C システムにおける対応するモジュール、装置、ユニットなどにより達成される。当業者であれば、前記システムの技術スキームに基づいて前記方法のステップを実現することができる。つまり、前記システムにおける実施例は、前記方法を実現するための好適例として理解され得るので、ここで説明を省略する。

10

【 0 0 9 7 】

また、本発明は、コンピュータ読み取り可能な記憶媒体に適用できる。コンピュータ読み取り可能な記憶媒体にコンピュータプログラムを記憶し、プロセッサによる実行により前記実施例に記載のメディア内容に基づく自己適応システムコード F E C 符号化および復号化方法のステップを実現する。

【 0 0 9 8 】

当業者に知られているように、ピュアコンピュータ可読プログラムコードの方式により本発明で提供されるシステムおよびその各装置、モジュール、ユニットを実現することに加えて、方法のステップを論理プログラミングすることにより、本発明で提供されるシステムおよびその各装置、モジュール、ユニットは、論理ゲート、スイッチ、特定用途向け集積回路、プログラマブルロジックコントローラおよび組み込みマイクロコントローラなどにより同じ機能を実現することができる。従って、本発明で提供されるシステムおよびその各装置、モジュール、ユニットは、ハードウェア部品と見なすことができ、それらの内部に含まれる各機能を達成するための装置、モジュール、ユニットは、ハードウェア部品内の構造と見なすことができる。さらに、各機能を達成するための装置、モジュール、ユニットは、方法を達成できるソフトウェアモジュールまたはハードウェア部品内の構造と見なすこともできる。

20

30

【 0 0 9 9 】

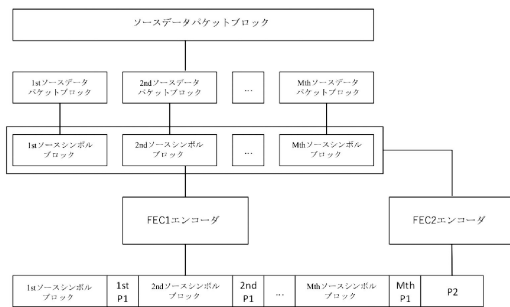
以上、本発明の具体的な実施例を説明した。需要理解的は、本発明は、前記特定の実施形態に限定されず、当業者は特許請求の範囲内において様々な変化または修正を行うことができ、これらの変化または修正は本発明の実質的な内容に提供を与えない。矛盾がない限り、本発明の実施例および実施例における特徴は、任意に組み合わせることができる。

40

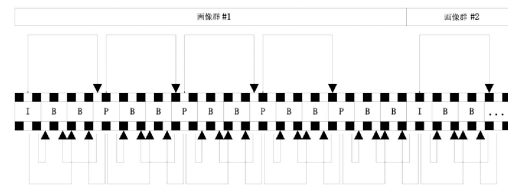
50

【図面】

【 図 1 】

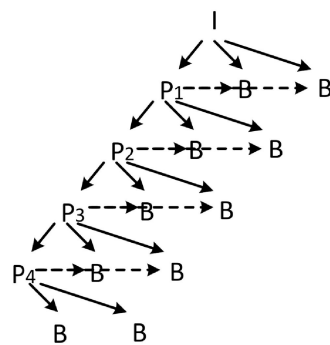


【圖 2】

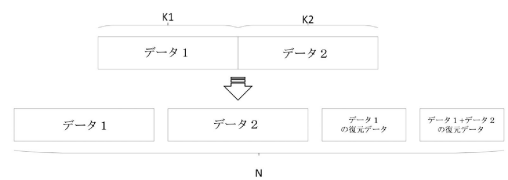


10

【 図 3 】

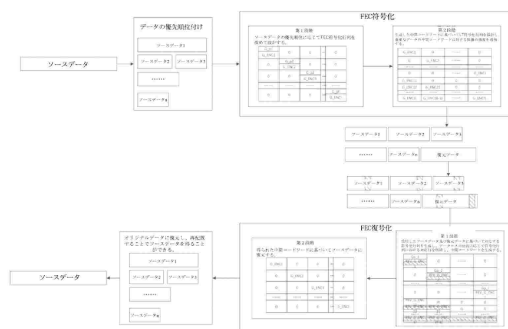


【 図 4 】

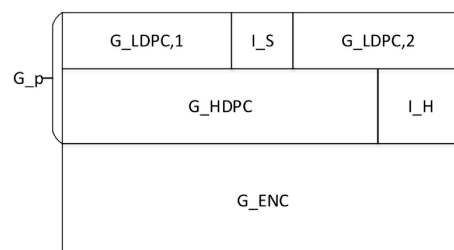


20

【 図 5 】



【 図 6 】



30

40

50

【図 7】

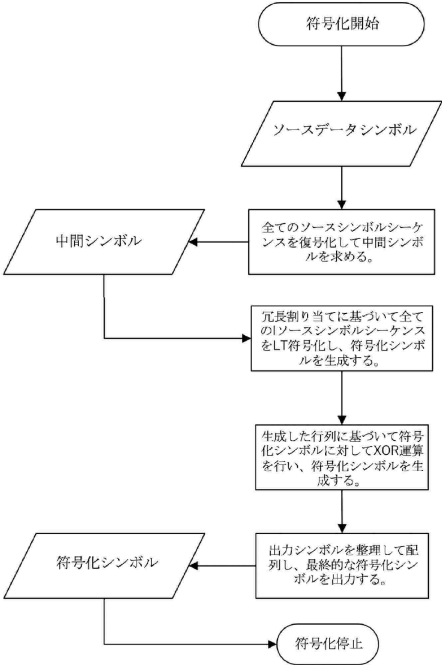
G_p	0
G_ENC1	
0	G_p
	G_ENC2

【図 8】

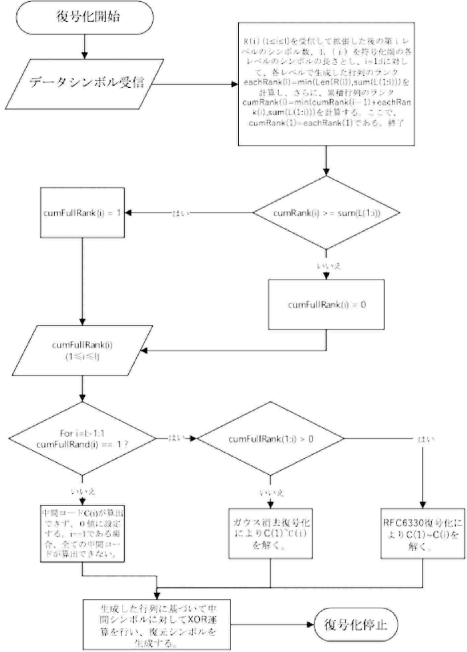
G_ENC1	0
0	G_ENC2
G_ENC1'	0
G_ENC1''	G_ENC2'

10

【図 9】



【図 10】



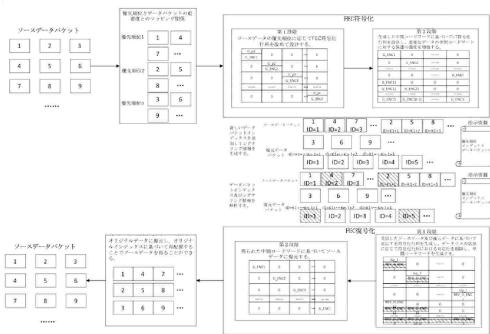
20

30

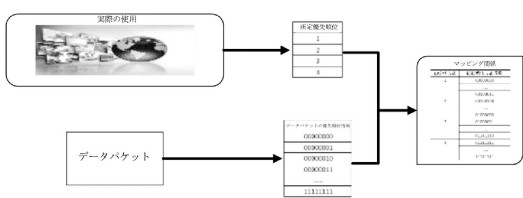
40

50

【図 1 1】

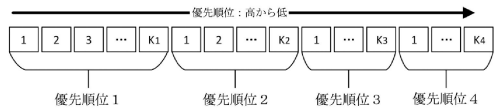


【図 1 2】

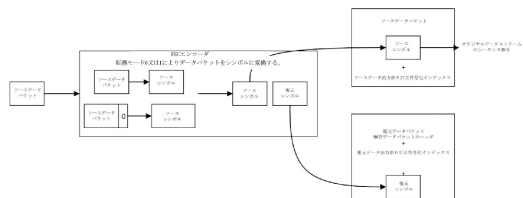


10

【図 1 3】

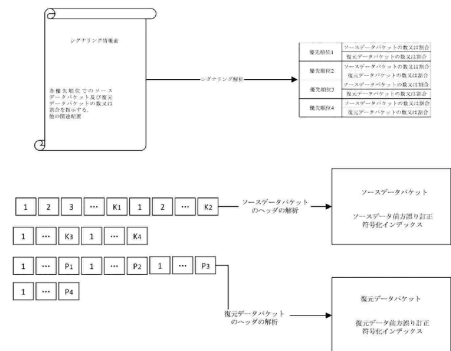


【図 1 4】

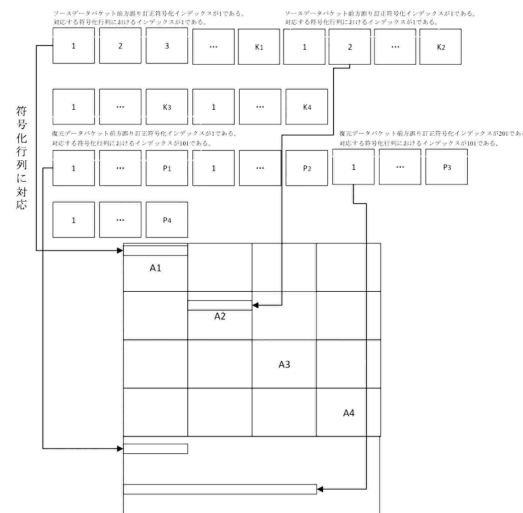


20

【図 1 5】



【図 1 6】



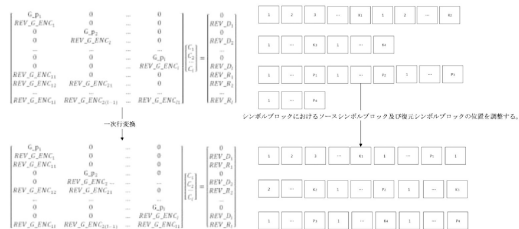
30

40

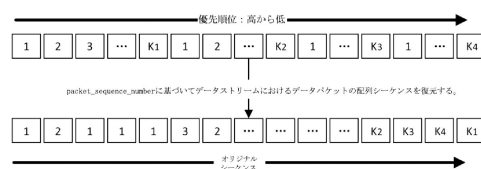
50



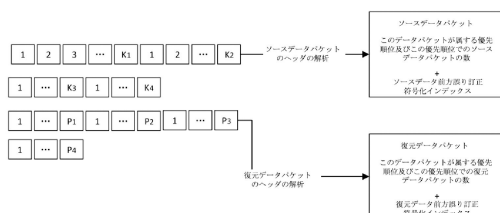
## 【 图 1 7 】



## 【圖 18】



## 【 図 19 】



10

20

30

40

50

## フロントページの続き

## (33)優先権主張国・地域又は機関

中国(CN)

## 前置審査

(72)発明者 張 文軍

中華人民共和国, 2 0 0 2 4 0, 上海市閔行区東川路 8 0 0 号

(72)発明者 柳 寧

中華人民共和国, 2 0 0 2 4 0, 上海市閔行区東川路 8 0 0 号

(72)発明者 管 云峰

中華人民共和国, 2 0 0 2 4 0, 上海市閔行区東川路 8 0 0 号

(72)発明者 王 延峰

中華人民共和国, 2 0 0 2 4 0, 上海市閔行区東川路 8 0 0 号

(72)発明者 孫 軍

中華人民共和国, 2 0 0 2 4 0, 上海市閔行区東川路 8 0 0 号

(72)発明者 姜 志乾

中華人民共和国, 2 0 0 2 4 0, 上海市閔行区東川路 8 0 0 号

(72)発明者 陳 浩

中華人民共和国, 2 0 0 2 4 0, 上海市閔行区東川路 8 0 0 号

審査官 吉江 一明

## (56)参考文献

Cornelius Hellge, Thomas Schierl, Thomas Wiegand, MOBILE TV USING SCALABLE VIDEO CODING AND LAYER-AWARE FORWARD ERROR CORRECTION, 2008 IEEE International Conference on Multimedia and Expo, IEEE, 2008年04月26日, pp. 1177-1180, <https://ieeexplore.ieee.org/document/4607650>

Michael Luby, Mark Watson, Tiago Gasiba, Thomas Stockhammer, Wen Xu, Raptor codes for reliable download delivery in wireless broadcast systems, CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006., IEEE, 2006年01月10日, <https://ieeexplore.ieee.org/document/1593014>

Cornelius Hellge, David Gomez-Barquero, Thomas Schierl, Thomas Wiegand, Layer-Aware Forward Error Correction for Mobile Broadcast of Layered Media, IEEE Transactions on Multimedia, Volume: 13, Issue: 3, IEEE, 2011年03月22日, pp. 551-562, <https://ieeexplore.ieee.org/document/5737793>

Qualcomm Incorporated, Proposed MBMS FEC Code: Proposed MBMS FEC Code based on IETF RFC6330[online], 3GPP TSG-SA WG4#70 S4-121053, 2012年08月10日

## (58)調査した分野 (Int.Cl., D B 名)

H 0 3 M 1 3 / 3 5

H 0 4 L 1 / 0 0

I E E E X p l o r e