

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第6523249号
(P6523249)

(45) 発行日 令和1年5月29日(2019.5.29)

(24) 登録日 令和1年5月10日(2019.5.10)

(51) Int.Cl.

H04L 29/06 (2006.01)
H03M 7/30 (2006.01)

F 1

H04L 13/00 305C
H03M 7/30 Z

請求項の数 16 (全 20 頁)

(21) 出願番号 特願2016-508247 (P2016-508247)
 (86) (22) 出願日 平成25年11月22日 (2013.11.22)
 (65) 公表番号 特表2016-519527 (P2016-519527A)
 (43) 公表日 平成28年6月30日 (2016.6.30)
 (86) 國際出願番号 PCT/IB2013/002615
 (87) 國際公開番号 WO2014/170715
 (87) 國際公開日 平成26年10月23日 (2014.10.23)
 審査請求日 平成28年11月17日 (2016.11.17)
 (31) 優先権主張番号 13305502
 (32) 優先日 平成25年4月17日 (2013.4.17)
 (33) 優先権主張国 歐州特許庁 (EP)

(73) 特許権者 501263810
 トムソン ライセンシング
 Thomson Licensing
 フランス国, 92130 イッシー レ
 ムーリノー, ル ジヤンヌ ダルク,
 1-5
 1-5, rue Jeanne d' A
 r c, 92130 ISSY LES
 MOULINEAUX, France
 (74) 代理人 100079108
 弁理士 稲葉 良幸
 (74) 代理人 100109346
 弁理士 大貫 敏史
 (74) 代理人 100117189
 弁理士 江口 昭彦

最終頁に続く

(54) 【発明の名称】パケットヘッダを圧縮する方法及び装置

(57) 【特許請求の範囲】

【請求項 1】

データを送信する方法であって、
 複数のトランスポートパケットを含むデータストリームに、ビデオデータとオーディオ
 データのうち少なくとも一方をカプセル化するステップであって、各トランスポートパケ
 ットはヘッダ部とペイロード部とを含むステップと、

フルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのう
 ち一方を用いて、あるトランスポートパケットを送信できるか判断するステップと、

判断されたフォーマットに応じて前記データストリームのトランスポートパケットを構
 成するステップであって、構成されるトランスポートパケットは、少なくとも、第1のヘ
 ッダを含む第1のパケットと、第2のヘッダを含む第2のパケットとを含み、前記第1の
 ヘッダは前記フルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは前記
 縮小サイズパケットヘッダフォーマットに対応するステップとを含み、

前記第1のパケットのシーケンス番号は前記第1のヘッダ中の複数のビットで表され、
 前記第2のパケットのシーケンス番号は、前記第2のパケットのシーケンス番号に対応
 する他の複数のビットの最下位の所定数のビットを用いて前記第2のヘッダ内に示され、

前記第2のヘッダ中の参照シーケンス番号は、前記複数のビットの一部で表され、前記
 第2のヘッダが前記第1のヘッダを参照として用いることを示すために用いられ、

前記方法は、

構成した前記トランスポートパケットを送信するステップとを含む、

10

20

方法。

【請求項 2】

前記構成するステップは、

前記第1のヘッダを有する前記第1のパケットを構成するステップと、

前記第2のヘッダを生成するステップであって、前記第2のヘッダは、送信を意図された値と、前記第1のヘッダ中の対応するフィールド中に表された値との間の差を表すフィールドを含む、ステップとを含む、

請求項1に記載の方法。

【請求項 3】

データを受信する方法であって、

10

複数のトランSPORTパケットを含み、ビデオデータとオーディオデータのうち少なくとも一方を含むデータストリームを受け取るステップであって、各トランSPORTパケットはヘッダ部とペイロード部とを含み、前記複数のトランSPORTパケットは少なくとも、第1のヘッダを有する第1のパケットと、第2のヘッダを有する第2のパケットとを含み、前記第1のヘッダはフルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは縮小サイズパケットヘッダフォーマットに対応する、ステップと、

前記第1のパケットのシーケンス番号は前記第1のヘッダ中の複数のビットで表され、

前記第2のパケットのシーケンス番号は、前記第2のパケットのシーケンス番号に対応する他の複数のビットの最下位の所定数のビットを用いて前記第2のヘッダ内に示され、

前記第2のヘッダ中の参照シーケンス番号は、前記複数のビットの一部で表され、

20

前記第1のヘッダのシーケンス番号と、前記第2のヘッダの参照シーケンス番号とに基づいて、前記第1のヘッダが前記第2のヘッダの参照として用いられるべきことを判断するステップと、

前記第1のヘッダを参照として用いて前記第2のヘッダを復号するステップとを含む、

方法。

【請求項 4】

前記受信するステップは、

前記第1のヘッダを含む前記第1のパケットを受信するステップと、

前記第2のヘッダを含む前記第2のパケットを受信するステップとを含み、

前記復号するステップは、

30

前記第2のヘッダ中のフィールドを、前記第1のヘッダ中の対応するフィールドに応じて復号するステップを含む、

請求項3に記載の方法。

【請求項 5】

データを送信する装置であって、

複数のトランSPORTパケットを含むデータストリームに、ビデオデータとオーディオデータのうち少なくとも一方をカプセル化するように構成され、各トランSPORTパケットはヘッダ部とペイロード部とを含む送信器と、

フルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのうち一方を用いてトランSPORTパケットを送信できるか決定し、前記決定されたフォーマットに応じて前記データストリームのトランSPORTパケットを構成するように構成されたヘッダ圧縮器とを有し、

40

構成されるトランSPORTパケットは、少なくとも、第1のヘッダを有する第1のパケットと、第2のヘッダを有する第2のパケットとを含み、前記第1のヘッダは前記フルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは前記縮小サイズパケットヘッダフォーマットに対応し、

前記第1のパケットのシーケンス番号は前記第1のヘッダ中の複数のビットで表され、

前記第2のパケットのシーケンス番号は、前記第2のパケットのシーケンス番号に対応する他の複数のビットの最下位の所定数のビットを用いて前記第2のヘッダ内に示され、

前記第2のヘッダ中の参照シーケンス番号は前記複数のビットの一部により表され、前

50

記第2のヘッダが前記第1のヘッダを参照として用いることを示すために用いられ、前記送信器は構成された前記トランSPORTパケットを送信するように構成される、装置。

【請求項6】

前記送信器は、前記第1のヘッダを有する前記第1のパケットを送信するように構成され、

前記ヘッダ圧縮器は前記第2のヘッダを生成するように構成され、前記第2のヘッダは、送信すべく意図された値と、前記第1のヘッダ中の対応するフィールドに表された値との差を表すフィールドを含み、前記送信器は前記第2のヘッダを含む前記第2のパケットを送信する、

10

請求項5に記載の装置。

【請求項7】

前記第2のヘッダ中の参照シーケンス番号は前記複数のビットの最下位の所定数のビットで表される、請求項1または3に記載の方法。

【請求項8】

前記第1のヘッダは、前記第2のヘッダ中のフィールドに表されるデータが、前記第2のヘッダ中のフィールドに含まれるビット数の最大表現能力を超えたときに構成される、

請求項1に記載の方法。

【請求項9】

データを受信する装置であつて、

20

複数のトランSPORTパケットを含み、ビデオデータとオーディオデータのうち少なくとも一方を含むデータストリームを受け取るように構成された受信器であつて、各トランSPORTパケットはヘッダ部とペイロード部とを含み、前記複数のトランSPORTパケットは少なくとも、第1のヘッダを有する第1のパケットと、第2のヘッダを有する第2のパケットとを含み、前記第1のヘッダはフルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは縮小サイズパケットヘッダフォーマットに対応する受信器と、

前記第1のパケットのシーケンス番号は前記第1のヘッダ中の複数のビットで表され、

前記第2のパケットのシーケンス番号は、前記第2のパケットのシーケンス番号に対応する他の複数のビットの最下位の所定数のビットを用いて前記第2のヘッダ内に示され、

前記第2のヘッダ中の参照シーケンス番号は、前記複数のビットの一部で表され、

30

前記第1のヘッダのシーケンス番号と、前記第2のヘッダの参照シーケンス番号とに基づいて、前記第1のヘッダが前記第2のヘッダの参照として用いられるべきことを判断するヘッダ復元器と、

前記第1のヘッダを参照として用いて前記第2のヘッダを復号するように構成される復号器とを有する、

装置。

【請求項10】

前記受信器は、前記第1のヘッダを含む前記第1のパケットを受信し、前記第2のヘッダを含む前記第2のパケットを受信し、前記ヘッダ復元器は、前記第2のヘッダ中のフィールドを前記第1のヘッダ中の対応するフィールドに応じて復号するように構成される、請求項9に記載の装置。

40

【請求項11】

前記第2のヘッダは、他の複数のビットの最下位の所定数のビットを用いて前記第2のパケットのシーケンス番号を示す第2のフィールドを含み、前記他の複数のビットは前記第2のパケットのシーケンス番号に対応する、

請求項1または3に記載の方法。

【請求項12】

前記第1のヘッダ中の対応するフィールドは一組のビットにより表され、前記第2のヘッダ中のフィールドは、前記一組のビットと、受信したフィールドとに応じて復号され、前記第2のヘッダ中の復号されたフィールドは、前記第1のヘッダ中の対応するフィール

50

ドを表すのに使われるのと同じビット数を有する、請求項 2 または 4 に記載の方法。

【請求項 13】

前記第 2 のヘッダは前記第 2 のヘッダが前記第 1 のヘッダを参照として用いることを示す前記複数のビットの最下位の所定数のビットを含む、

請求項 5 または 9 に記載の装置。

【請求項 14】

前記第 2 のヘッダは前記第 1 のパケット中の第 1 のヘッダより少ないフィールドを含む

、
請求項 5 または 9 に記載の装置。

【請求項 15】

10

前記第 2 のパケット中の第 2 のヘッダは前記第 1 のパケット中の第 1 のヘッダより少ないフィールドを含む、請求項 1 または 3 に記載の方法。

【請求項 16】

コンピュータに、請求項 1 - 4、7、8、11、12 及び 15 いずれか一項に記載の方法のステップを実行させるコンピュータプログラム。

【発明の詳細な説明】

【技術分野】

【0001】

【関連出願との相互参照】

本出願は、2013年5月13日出願、発明の名称「Method and Apparatus for Packet Header Compression」の欧洲特許出願第EP13305502.0号の出願日の利益を主張するものである。この出願はその全体をここに参照援用する。

20

【0002】

本発明は、圧縮トランSPORTパケットヘッダを生成する方法及び装置と、圧縮トランSPORTパケットヘッダを処理する方法及び装置とに関する。

【背景技術】

【0003】

通信ネットワークにおいて、最も費用のかかるリソースのひとつは回線容量である。通信ネットワークを通じて伝送されるデータ量を効果的に削減するオーディオ及びビデオの圧縮方法が開発されている。また、トランSPORTプロトコルの利用により生じるオーバーヘッドを削減することが重要である。より具体的に、トランSPORTパケットヘッダのサイズを縮小することが重要である。

30

【発明の概要】

【0004】

本原理は、データを送信する方法を提供する。該方法は、複数のトランSPORTパケットを含むデータストリームに、ビデオデータとオーディオデータのうち少なくとも一方をカプセル化するステップであって、各トランSPORTパケットはヘッダ部とペイロード部とを含むステップと、フルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのうち一方を用いて、あるトランSPORTパケットを送信できるか判断するステップと、前記判断するステップに応じて前記データストリームのトランSPORTパケットを構成するステップと、上記の通り構成された前記トランSPORTパケットを送信するステップとを含む。本原理は、これらのステップを実行する装置も提供する。

40

【0005】

本原理は、データを受信する方法も提供する。該方法は、複数のトランSPORTパケットを含み、ビデオデータとオーディオデータのうち少なくとも一方を含むデータストリームを受け取るステップであって、各トランSPORTパケットはヘッダ部とペイロード部とを含むステップと、トランSPORTパケットがフルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのうち一方を用いているか判断するステップと、上記の通り判断された前記フォーマットに応じて前記データストリームのトランSPORT

50

パケットを復号するステップと、を含む。本原理は、これらのステップを実行する装置も提供する。

【0006】

本原理は、上記の方法によりデータを送信または受信する命令を記憶したコンピュータ読み取り可能媒体も提供する。

【図面の簡単な説明】

【0007】

【図1A】ペイロードデータに先行するMMT (MPEG Multimedia Transport) ペイロードヘッダの構造を示す図である。

【図1B】MMTペイロードヘッダエクステンションの構造を示す図である。

【図2】MMTパケットヘッダの構造を示す図である。

【図3】本原理の一実施形態による、縮小ヘッダ生成方法の一例を示すフロー図である。

【図4】本原理の一実施形態による、フルサイズヘッダのシーケンス番号中のビットを用いて縮小ヘッダの reduced_SeqNum を生成すること、及び reduced_SeqNum を用いて縮小ヘッダのシーケンス番号を復号することを示す図である。

【図5A】フルヘッダを有する修正MMTパケットの構造を示す図である。

【図5B】本原理の一実施形態による縮小ヘッダを有する修正MMTパケットの構造を示す図である。

【図6A】ペイロードデータに先行する修正MMTペイロードヘッダの構造を示す図である。

【図6B】本原理の一実施形態によるペイロードデータに先行する縮小MMTペイロードヘッダの構造を示す図である。

【図7】本原理の一実施形態による、フルヘッダを用いるか縮小ヘッダを用いるか判断する方法の一例を示すフロー図である。

【図8】本原理の一実施形態による、縮小ヘッダ復元方法の一例を示すフロー図である。

【図9】本原理の一実施形態による送信システムの一例を示すブロック図である。

【図10】本原理の一実施形態による受信システムの一例を示すブロック図である。

【図11】本原理の一実施形態による受信システムの他の一例を示すブロック図である。

【発明を実施するための形態】

【0008】

MPEG-Hパート1標準 (MPEG Multimedia Transport すなわち MMT としても知られている) は、タイムド・メディア・コンテンツ及びノンタイムド・メディア・コンテンツのパッケージング、伝送及び構成に対する完全なソリューションを規定している。MMTは目下開発中であり、標準案は「Text of ISO/IEC 2nd CD 23008-1 MPEG Media Transport, MPEG/N13293」(Geneva, Switzerland, January 2013 (以下、「MMT_CD」と呼ぶ))に記載されている。

【0009】

MMTは主にIPネットワークに関するものであるが、任意のタイプのパケットベースネットワークによるコンテンツ配信もサポートしている。具体的に、MMTは、地上波、ケーブル又は衛星ネットワークなどのブロードキャストネットワークによるオーディオビジュアルサービスの配信に用いられ得る。

【0010】

MMTでは、「アセット (Asset)」との用語は、同じトランスポート特性を有するデータを含み、同じアセットID (Asset ID) を有する一以上のMPU (メディア処理ユニット) により構成されたデータエンティティを指す。「パッケージ (Package)」との用語はデータの論理的コレクションを指し、一以上のアセット及びそれに関係するアセット配信特性 (Asset Delivery Characteristics) (すなわち、アセットの配信に必要なサービス品質 (Quality of Service) に関する記述) 及び構成情報 (Composition Information)

20

30

40

50

ation) (すなわち、アセット間の空間的及び時間的関係に関する記述)により構成されている。

【0011】

MMTパッケージング及びトランスポート機能は、2つのプロトコルで、すなわちMMT-Payload Format (MMT-PF)とMMT-Transport Protocol (MMT-TP)とで規定されている。具体的に、MMT-PFはマルチメディアパッケージのコンテンツ成分(例えば、オーディオ、ビデオ及びファイル)をパケット化する一般的ペイロードフォーマットを規定している。MMT-PFはメディアデータを符号化するのに用いる具体的なメディアコードにはとらわれず、シグナリングメッセージと順方向誤り訂正(Forward Error Correction (FEC))情報をパケット化するのにも使われる。MMTペイロードフォーマットは、 RTPまたはMMTトランスポートプロトコルなどの任意のパケットベースのトランスポートプロトコルに対して利用できる。MMT-TPは、IPネットワーク環境を含むパケットベースの異種配信ネットワークによるパッケージのストリーミング配信をサポートするトランスポートプロトコルを確定する。MMTプロトコルは、パッケージ(Package)の配信のための基本的機能を提供する。例えば、さまざまなアセットが単一のMMTパケットフローにより配信できるようにするプロトコルレベル多重化、広い範囲のネットワークジッターに適合するプレゼンテーション時間に依存しない配信タイミングモデル、及びサービス品質(Quality of Service)をサポートする情報を提供する。

10

20

【0012】

図1Aと図1Bはそれぞれ、MMT-PFプロトコルによるペイロードデータに先行するMMTペイロードヘッダの構造と、MMTペイロードヘッダエクステンションの構造とを示している。このアプリケーションでは、ペイロードヘッダはMMT-PFヘッダとも呼ばれる。セマンティックスの完全な説明はMMT_CDに記載されている。図1Aと図1Bに示した幾つかのフィールドのセマンティックスを以下に再掲するが、これらは本原理により削除しても修正してもよい。

`length (16 bits)` - このフィールドは、ヘッダを除くペイロードの長さをバイト単位で示している。これはパディングデータ(padding data)のサイズを含む。

30

`f_i (2 bits)` - このフィールドは、フラグメンテーションインジケータがペイロード中のデータユニットのフラグメンテーション(fragmentation)に関する情報を含むことを示している。

`fragmentation_flag (F:1 bit)` - `fragment_counter`があれば「1」に設定される。

`aggregation_flag (A:1 bit)` - `aggregation_info`があれば「1」に設定される。

`RAP_flag (R:1 bit)` - ペイロードがランダムアクセスポイント(又はその一部)を含めれば「1」に設定される。

`payload_sequence_flag (P:1 bit)` - `payload_sequence_number`があれば「1」に設定される。

40

`number_data_unit (numDU:4 bits)` - このフィールドはこのMMTペイロード中のデータユニット数を示す。`fragmentation_flag`が「1」に設定されたとき、このフィールドは「0」である。

`DU_offset (16 bits)` - このフィールドは`data_offset`により示される、バイトの各データユニットの位置を示す。`aggregation_flag`が「1」に設定されたとき、このフィールドが使われる。

`payload_sequence_number (32 bits)` - このフィールドは同じアセットに関連するペイロードのシーケンス番号を示す。

【0013】

50

図2は、MMT-TPプロトコルによるMMTパケットヘッダの構成を示している。セマンティックスの完全な説明はMMT_CDに記載されている。本願では、MMTパケットヘッダはMMT-TPヘッダとも呼ぶ。図2に示した幾つかのフィールドのセマンティックスを以下に再掲するが、これらは本原理により削除しても修正してもよい。

packet_id(16 bits) - このフィールドは、あるアセットを他のアセットから区別するために各アセットに割り当てられた整数値である。

packet_sequence_number(32 bits) - このフィールドは、packet_idによりスコープされ、任意の値で始まりMMTパケットごとに1ずつインクリメントされる整数値である。最大値に達した後、「0」に戻る。

timestamp(32 bits) - このフィールドはMMTパケット配信の時間インスタンスを指定する。NTP(Network Time Protocol)時間は、IETF RFC5905, NTP version 4のクローズ6の「ショートフォーマット(short-format)」に記載されているように、タイムスタンプで使われる。このタイムスタンプはMMTパケットの第1ビットで測定される。 10

【0014】

MMT-PFとMMT-TPは両方とも、可変サイズヘッダを含み、(普通にシーケンス番号を有する)MMT-PFの場合に最小サイズの9バイトであり、MMT-TPの場合に12バイト(及び3ビット)である。MMT-PFとMMT-TPを用いて非常に小さいパケットペイロードを伝送してもよいので、ヘッダに費やされるこれらの余分な21バイトは非常に高いオーバーヘッドを表しているかも知れない。 21

【0015】

ヘッダのサイズを縮小するため、Robust Header Compression(RFC3095で定義されたRoHC)を用いても良い。RoHCは、ヘッダのサイズを効果的に縮小できるが、受信側で大量の計算を要する可能性がある複雑なコーディング手法に依存し、受信器がパケットフィルタリングを目的としてヘッダの一部を調べればよい場合でも、受信器にすべてのヘッダを復号することを強制する。

【0016】

本原理は、上記の問題を解決し、少ない計算の複雑性でヘッダサイズを大幅に縮小できる方法と装置を提供する。

【0017】

一実施形態では、ヘッダは圧縮して送信しても、圧縮せずに送信してもよい。本願では、圧縮していないヘッダは「フルサイズヘッダ」または「フルヘッダ」と呼び、圧縮されたヘッダは「縮小サイズヘッダ」または「縮小ヘッダ」と呼ぶ。ヘッダ圧縮が使われているか区別するため、フィールド(本実施形態ではCフィールドとする)を、例えばヘッダの始めに付け加えることができる。一例では、Cが0に設定されている場合、ヘッダは「フルサイズ(full size)」ヘッダを含み、Cが1に設定されている場合、ヘッダは「縮小サイズ(reduced size)」ヘッダを含む。 30

【0018】

縮小サイズヘッダを構成するため、フルサイズヘッダの幾つかのフィールドを削除してもよく、幾つかのフィールドをフルヘッダより少ないビットで表してもよく、新しいフィールドを追加してもよい。ヘッダ中のフィールドの順序を変更してもよい。以下、MMT-TPとMMP-PFの場合のヘッダ圧縮の実施形態をさらに詳しく説明する。 40

MMT-TPヘッダ

図3は、縮小MMT-TPヘッダを生成する方法例300を示す。方法300はステップ305から始まる。ステップ310において、delta_timestampフィールドを用いてタイムスタンプを表す。多くのビットを要するが、パケットに関連するタイムスタンプ情報は保存する必要がある。

【0019】

delta_timestampフィールドは、参照フルサイズヘッダのタイムスタン

プフィールドと、フルヘッダを使った場合に現在のタイムスタンプフィールドに入るだろ
う値との間の差を含む。一実施形態では、フルサイズヘッダは、受信器と送信器の両方が
知っている規則に基づく参照ヘッダであることを暗示してもよく、例えば最後に受信した
フルサイズヘッダを参照ヘッダとして用いてもよい。他の一実施形態では、フルサイズヘ
ッダが参照ヘッダであると明示的に示されてもよい。タイムスタンプ間の差は、NTP
タイムスタンプの最下位から19個のビットと同様の方法で符号化される。これは、8秒の
長さのタイムスタンプ精度を保つ。これらの2つのタイムスタンプの間の差が8秒より長
い場合（そのため19ビットで符号化できる最長時間を超える場合）、縮小サイズヘッダ
を有する別のパケットに新しいタイムスタンプ参照値を提供するために、フルヘッダを有
するパケットが送信される。

10

【0020】

フルヘッダのpacket_idフィールドは、reduced_packet_idフィールドで置き換えられる。reduced_packet_idはフルヘッダにおける16ビットではなく8ビットを用いる。それゆえ、この16ビットパケットIDから8ビットIDへの縮小は、packet_idが0ないし255であるストリームに対する縮小ヘッダの使用を制限する。それゆえ、ステップ320において、packet_idが0と255との間にあるかチェックする。YESであれば、方法300はreduced_packet_idの値を設定し、続けて縮小ヘッダを生成する。NOであれば、提案のヘッダ縮小メカニズムは使用できず、フルヘッダを生成しなければならない。

【0021】

ステップ330において、reduced_SeqNumフィールドを用いてパケットのシーケンス番号を表す。reduced_SeqNumフィールドは、フルサイズヘッダが使われたとしたらヘッダの中にあるpacket_sequence_numberフィールドの最下位から8つのビットを含む。この新しいフィールドは8ビットで符号化されているので、本原理は、（同じpacket_idで特定される）各ストリームに対して、フルサイズヘッダを有するパケットが少なくとも256パケットごとに送信させる。

20

【0022】

ステップ340において、新しいフィールドRefSeqNumを生成する。RefSeqNumは、フルヘッダが参照として用いられるパケットのシーケンス番号の少なくとも5ビットを含む。この新しいフィールドにより、受信器が、最後に受信したフルヘッダが現在の縮小サイズヘッダの参照として使われるものかチェックできるので、さらにロバストになる。パケットは失われることもあるので、フィールドRefSeqNumは、フルヘッダ参照が受信されていない場合に受信器が縮小ヘッダを不適切に復号することを試みないようにするメカニズムを提供する。

30

【0023】

ステップ350において、「delta_timestamp」、「reduced_packet_id」、「reduced_SeqNum」、「RefSeqNum」その他のフィールドが、例えば図5Bに示したフォーマットにより、縮小サイズヘッダ(reduced_size_header)に書き込まれる。方法300はステップ399で終わる。

40

方法300のステップは、図3に示した順序とは異なる順序で進行してもよく、例えばステップ310-340はどんな順序で行ってもよい。

【0024】

図4は、フィールド「reduced_SeqNum」を生成するためシーケンス番号中のビットが用いられること、及び縮小ヘッダ中のシーケンス番号を復号するためにフィールド「reduced_SeqNum」を用いられることを、例を用いて示している。この例では、現在のパケットN+iが送信され、パケットNで最後のフルサイズヘッダが受信される。送信側では、シーケンス番号(415)の最下位から8つのビット(430)のみを保存して、フィールド「reduced_SeqNum」を生成する。これによ

50

り必要な処理が削減される。

【0025】

受信側における復号は送信側における符号化よりも複雑である。まず、逆圧縮器 (de-compressor) が、参照として用いられるパケットの、シーケンス番号 (405)、またはシーケンス番号の最上位から 24 個のビット (410) を記憶する必要がある。逆圧縮器は、reduced_SequenceNum の初期値として、シーケンス番号の最下位から 8 つのビット (430) を記憶することとしてもよい。次に、受信した各縮小ヘッダについて、復号器は、reduced_SequenceNum が 0 を通ってループしたか追跡する必要がある。縮小ヘッダパケットの完全なシーケンス番号は、reduced_SequenceNum (430) を、参照シーケンス番号の最上位から 24 ビット (440) (reduced_SequenceNum が 0 を通ってループしたとき 1 だけインクリメントされる) に附加することにより得られる。すなわち、440 に示されたビットは、reduced_SequenceNum が 0 を通ってループしていなければ、410 に示されたビットに等しく、そうでなければ、410 に示されたビットより 1 だけ大きい。

【0026】

図 5 A は提案のフル MMT - TP ヘッダを示し、図 5 B は、本原理による提案の縮小 MMT - TP ヘッダを示す。図 2 に示した MMT - TP ヘッダと比較して、フィールド「Q」、「F」、「P」、「FEC」、「RES」、「TB」、「DS」及び「R」が提案のフル MMT - TP ヘッダの始めに移動され、そのサイズとセマンティックス (semantics) は変更されておらず、フィールド「S」が削除され、フィールド「C」と「I」が追加されている。現在のヘッダ情報が後で参照として使われる所以それを記憶すべきか示すために「I」フラグが用いられ、I フラグはフル MMT - PF 及び MMT - TP ヘッダに追加されている。

【0027】

フィールドの順序も調整されている。フラグ「C」は、用いられるヘッダの種類を示すので、復号器により決定される最初の情報であることを要する。復号器が最初のビットから決定すると過程すると、フラグ「C」に最初のビットを用いることにより、復号器は続くヘッダの種類を最初に決定できる。

【0028】

図 5 B に示した提案の縮小 MMT - TP ヘッダを図 5 A に示したフル MMT - TP ヘッダと比較すると、「packet_id」、「packet_sequence_number」及び「timestamp」がそれぞれ「reduced_packet_id」、「reduced_SequenceNum」及び「delta_timestamp」と置き換えられており、フィールド「RefSeqNum」が追加されており、フィールド「I」、「RES」及び「reserved」が削除されている。その結果、最小の MMT - TP ヘッダのサイズは 99 ビットから 56 ビットに減少し、これはビットの 43 % 節約である。

【0029】

MMT - TP ストリームは、その packet_id により特定され、各ストリームはそれ自体の packet_sequence_number シーケンスを有する。それゆえ、(packet_id が 0 ないし 255 の) 限定された数のストリームにヘッダ縮小メカニズム (header reduction mechanism) を適用することにより、RefSeqNum と reduced_packet_id フィールドの組み合わせにより、縮小サイズヘッダの復号に用いる参照パケットを独特な方法で特定する。

【0030】

さらに、reduced_packet_id は packet_id フィールドの可能な値の範囲がより小さいコピーでしかないので、packet_id のフィルタリングによりなされる従来のパケットフィルタリングは、reduced_packet_id に同様に使える。一例では、フィルタリングは、パケットを送信または受信すべきか決定するために、幾つかのフィールドを調べることである。ヘッダの一フィールドを読むために R0HC などの手法を用いる場合、ヘッダ全体を復元する必要がある。対照的に、提案の手法を使

えば、所望のフィールドのみを復元すればよい。すなわち、フィルタリングは、元の `packet_id` を再構成せずに、`reduced_packet_id` に直接的に行える。それゆえ、本原理によるヘッダ圧縮手法はパケットフィルタリングに対して完全にトランスペレント (`transparent`) である。

MMT-PF ヘッダ

通常、ヘッダ圧縮は、大きくかつ繰り返されるヘッダに対して関心を寄せる。MMT の場合、FEC リペアシンボルの場合に、または単一の MPU が複数のセグメントで伝送されたときに、ヘッダ圧縮により大幅なビット節約が提供される。このように、ヘッダ圧縮により利益を受ける幾つかの具体的な使用シナリオに対して、縮小ヘッダを設計する。使用シナリオを検討して、下記の縮小サイズ MMT-PF ヘッダを提案する。 10

【0031】

(1) アグリゲーション (`aggregation`) は縮小サイズヘッダによりサポートされていないので、`A` と `number_data_unit` と `DU_offset` フィールドを削除した。その結果、アグリゲーションを使うとき、フルヘッダを使用する。

【0032】

(2) ペイロード中に `Random Access Point (RAP)` がある場合、常にフルヘッダが使われるので、`R` フラグは削除した。これにより、RAP を有するパケットはそれ自体で（他の「参照」パケットの符号化に依存せずに）復号できるようになる。 20

【0033】

(3) 縮小ヘッダを有するすべてのパケットはその「参照」パケットと同じ `payload_ad_sequence_number` を有するので、`P` 及び `payload_sequence_number` フィールドは削除した。

【0034】

(4) ヘッダ圧縮は同サイズのフラグメントにのみ使われるので、`length` フィールドを削除した。その結果、最初のフラグメントはフルヘッダを用い、「参照」フラグメント（通常は最初のフラグメントであるが、必ずしもそうではないこともある）とは異なるサイズを有するフラグメントもフルヘッダを用いる。

【0035】

(5) `RefSNum` フィールドは、フルヘッダが参照として用いられるパケットのペイロードシーケンス番号の少なくとも 4 ビットを含む。これにより、受信器が、最後に受信したフルヘッダが現在の縮小サイズヘッダの参照として使われるものかチェックできるので、さらにロバストになる。パケットは失われることもあるので、これは、フルヘッダ参照が受信されていない場合に受信器が縮小ヘッダを不適切にデコードすることを試みないようとするメカニズムを提供する。 30

【0036】

図 6 A は提案のフル MMT-PF ヘッダを示し、図 6 B は提案の縮小 MMT-PF ヘッダを示す。図 1 A に示した MMT-PF ヘッダと比較して、フィールド「`f_i`」、「`A`」、「`R`」、「`P`」及び「`E`」が、提案のフル MMT-PF ヘッダの始めに移動され、そのサイズとセマンティックスは変更していない。フラグ「`F`」と「`S`」が削除され、フィールド「`C`」と「`I`」が追加されている。フィールドの順序も調整されている。図 6 B に示した提案の縮小 MMT-PF ヘッダを図 6 A に示したフル MMT-PF ヘッダと比較すると、フィールド「`I`」、「`A`」、「`R`」、「`P`」、「`length`」、「`numDU`」、「`DU_offset`」及び「`payload_sequence_number`」が削除され、フィールド「`RefSNum`」が追加されている。その結果、最小の MMT-PF ヘッダのサイズは 128 ビットから 32 ビットに減少し、これはビットの 75 % 節約である。 40

【0037】

多くのアプリケーションにおいて、同じアセットの最後のフラグメント以外のすべての

フラグメントは、同じサイズを有し、それゆえ最初と最後のフラグメントのみでフルサイズヘッダを使え、他のすべてのフラグメントは、最初のフラグメントを参照として使いつつ、縮小ヘッダを使ってもよい。

【0038】

上記の通り、MMT-PFとMMT-TPに対して、異なる方法を用いて、効率的かつロバストなヘッダ圧縮を提供した。一実施形態では、現在のフィールドと参照フィールドとの間の差、例えば `delta_timestamp` を縮小ヘッダで用いる。フィールド自体ではなく差を用いることにより、より少ないビットを用いてフィールドを表せる。他の一実施形態では、現在のフィールドの下位ビット、例えば `reduced_SeqNum` を用いて現在のフィールドを表す。かかるフィールドを受信したとき、そのフィールドを復元するため参照フィールドの上位ビットが必要である。これらの2つの実施形態は、すなわち、差を用いるものと上位ビットを用いるものは、現在のフィールドの異なる表現を用いてヘッダ圧縮を効率化する。他の一実施形態では、本原理による構成は、ヘッダ圧縮の利益がある典型的な使用シナリオを判断し、さらに縮小ヘッダにおいて幾つかのフィールドを削除できると判断する。また、より少ないビットを用いて縮小ヘッダ中の現在のフィールドを表す（例えば、`reduced_packet_id`）。これにより、現在のフィールドにより表され得る値に制約がかかることがある。本原理は、縮小ヘッダ使用の制約を認め、フィールドの値を設定するときのルールとガイドラインとを提供する。

【0039】

上記の通り、縮小ヘッダの使用にはルールと制約とがある。図7は、ルールと制約とを検討して縮小ヘッダまたはフルサイズヘッダを用いるか判断する方法例700を示す。この方法は、例えば、MMT-TPによるトランスポートストリームに関連するヘッダを符号化する符号化器で実行される。方法700はステップ710で始まり、初期化を行う。

【0040】

ステップ720において、現在のパケットのタイムスタンプと、参照パケットのタイムスタンプとの間の差が8秒より大きいか（及びそれゆえ19ビットの `delta_timestamp` フィールドで符号化できないか）チェックする。YESであれば、方法700はステップ770においてフルサイズヘッダを生成する。

【0041】

ステップ730において、`packet_id` が0ないし255の範囲にあるかチェックする。NOであれば、`packet_id` は `reduced_packet_id` により正しく表せる範囲を超えており、ステップ770においてフルサイズヘッダが生成される。

【0042】

ステップ740において、（`packet_id` により特定される）各ストリームに対して、`reduced_SeqNum` がその初期値で終わるか、チェックする。YESであれば、方法700はステップ770においてフルサイズヘッダを生成して、生成したヘッダを用いるさらに別のパケットに対して、参照シーケンス番号を提供する。

【0043】

ステップ750において、パケットが、Random Access Point (RAP) を含むアクセスユニットの最初のパケットであるかチェックする。YESであれば、方法700はステップ770においてフルサイズヘッダを生成する。そうでなければ、ステップ760において、縮小ヘッダを生成する。

【0044】

方法700のステップは、図7に示した順序とは異なる順序で進行してもよく、例えばステップ720-750はどんな順序で行ってもよい。方法700では、その他の条件の下で、例えばユーザ要求に基づき、フルサイズヘッダを生成することを選択してもよい。MMT-PFに対してフルサイズヘッダまたは縮小ヘッダを使えるか判断するため、次の条件を検討し得る：

(1) アグリゲーションメカニズム (aggregation mechanism) が使われている；

10

20

30

40

50

(2) Random Access Point (RAP) がある；

(3) 新しいペイロードシーケンス番号が使われている；

(4) 現在のパケットの length 値が参照パケットの length 値と異なる。

【0045】

上記の条件のうち少なくとも一つが満たされた場合、フルMMT-PFヘッダを送信すべきである。他の条件の下で、例えばユーザ要求に基づき、フルMMT-PFヘッダを送信することもできる。

受信側において、本原理による方法を実施する受信器は、例えば「C」フラグを用いて、フルサイズヘッダが用いられているかまたは縮小ヘッダが用いられているか判断する。受信器は、フルヘッダが参照としてマークされているとき、フルヘッダから得た重要な情報を、例えばヘッダの復元に必要なものを記憶する。

【0046】

図8は、縮小サイズMMT-PFヘッダを復元する方法例800を示す。方法800はステップ805から始まる。ステップ810において、縮小ヘッダを受信し、縮小サイズヘッダから得たフィールド、例えば限定ではないが「delta_timestamp」、「reduced_SeqNum」及び「RefSeqNum」などのフィールドを解析する。ステップ820において、参照として使われるフルサイズヘッダのpacket_sequence_numberの少なくとも5ビットがRefSeqNumと同じであるかチェックする。それらが同じでなければ、縮小サイズヘッダは正しく復号できず、方法800はステップ899で終了する。そうでなければ、ステップ830において、現在の縮小サイズヘッダの参照として用いられているフルサイズヘッダのタイムスタンプとpacket_sequence_numberを決定する。ステップ840において、参照フルヘッダのタイムスタンプと、delta_timestampの和として、現在のパケットのタイムスタンプを決定する。ステップ850において、例えば図4に示したように、フルサイズヘッダのpacket_sequence_numberとreduced_SeqNumとに基づいて、現在のパケットのパケットシーケンス番号を決定する。方法800はステップ899で終わる。

【0047】

MMT-PFとMMT-TPヘッダを例として用いて、さまざまな実施形態をどのように用いて効率的にヘッダを圧縮するか説明する。ヘッダ圧縮なので、現在のフィールドを正しく復号するために正しい参照フィールドを用いることが重要である。復号をロバストにするため、参照フィールド（例えば、RefSeqNum）へのリンクを用いて、縮小サイズヘッダの間違った復号を防止できる。本原理は、他のアプリケーションやプロトコルにおいてヘッダ圧縮に用いることもできる。

【0048】

上記のさまざまな例では、フィールドの具体的な値（例えばフィールドやフラグのビット数など）や具体的な順序を説明した。これらの値や順序は、本原理が異なるアプリケーションやトランスポートプロトコルに適用される時に適宜調整する必要があるかも知れない。

【0049】

本原理による縮小ヘッダにおいて符号化されたパケットは、すべてのパケットの複雑なヘッダ復号をしなくても容易にフィルタできる。このように、本原理はネットワークの透明性という利点を提供する。また、単純なヘッダ圧縮メカニズムを用いることにより、本原理は、低い計算複雑性で実装でき、受信器において大がかりな処理を必要としない。

【0050】

図9は、送信システム例900を示す。入力データは、例えばオーディオデータとビデオデータであるが、メディア符号化器910において符号化される。符号化されたデータは多重化器920で多重化され、送信器930で送信される。例えば方法300と700である本原理によるヘッダ圧縮メカニズムは、多重化器920または送信器930に配置されたヘッダ圧縮器（940、950）で用いることができる。送信システムは、帯域幅

10

20

30

40

50

が高価な資源である一般的な放送テレビジョン環境において用いてもよいし、またはオーディオビジュアルサービスを提供するモバイルデバイスで用いてもよい。メディア符号化プロセスに続いて、多重化器 920においてヘッダ圧縮を用いることにより、予めヘッダ圧縮を準備することができる。MMT-PF フローは実際に送信される前にファイルフォーマットで記憶されてもよいからである。送信器 930において、本システムは、MMT-PF ヘッダ圧縮を（場合によっては更新して）再利用してもよく、実際の MMT-TP パケットを送信する前に MMT-TP ヘッダ圧縮を利用してよい。

【0051】

図 10 は、受信システム例 1000 を示す。システム 1000 の入力データは、トランスポートビットストリームであってもよく、例えばシステム 900 の出力であってもよい。データは、受信器 1010 で受信され、逆多重化器 1020 で逆多重化され、メディア復号器 1030 で復号される。MMT-TP パケットを受信したとき、ヘッダ復元器（1040、1050）が、参照パケットから得たヘッダ情報を記憶して、縮小ヘッダを復号することにより、MMT-TP ヘッダ復元を行う。復号されたパケットは逆多重化器 1020 のバッファに入れておける。MMT-PF ペイロードを処理するとき、逆多重化器 1020 は、MMT-PF ヘッダ復元を用いてよい。

【0052】

図 11 は、他の受信システム例 1100 を示す。概して、図 11 のビデオ受信システムにおいて、放送番組コンテンツを表すオーディオ、ビデオ及び関連データを担う信号で変調された放送キャリアが、アンテナ 10 で受信され、ユニット 13 で処理される。得られたデジタル出力信号は復調器 15 により復調される。ユニット 15 からの復調出力は、復号器 17 により、トレリス復号され、バイト長データセグメントにマッピングされ、逆インターリーブされ、リードソロモンエラー訂正される。ユニット 17 の出力データは、MPEG 互換トランスポート・データストリームの形式であり、例えば番組を代表する多重化されたオーディオ、ビデオ及びデータコンポーネントを含む MMT トランスポートストリームの形式である。ユニット 17 から出力されるトランスポートストリームは、ユニット 22 によりオーディオ、ビデオ及びデータコンポーネントに逆多重（demultiplex）され、これらは復号器 100 の他の要素によりさらに処理される。

【0053】

例えば MMT-PF 及び MMT-TP ヘッダ圧縮などのヘッダ圧縮手法が使われる場合、ユニット 22 は、ストリームを逆多重し、エレメンタリストリームをユニット 25、35 または 95 に送信する前に、ヘッダ復元を行う。一モードでは、復号器 100 は、ユニット 50 と 55 でそれぞれ表示及びオーディオ再生する復号された MPEG データを提供する。他の一モードでは、ユニット 17 から出力されるトランスポートストリームは、復号器 100 により処理され、記憶デバイス 90 を介して記憶媒体 105 に記憶される MPEG 互換データストリームを提供する。

【0054】

ユーザは、リモートコントロールユニット 70 を用いて、テレビチャンネルまたは番組ガイドなどのオンスクリーンメニューを見る選択をする。プロセッサ 60 は、視聴する所望の番組チャンネルを受信するように図 11 の要素を適切に設定する、インタフェース 65 を介してリモートコントロールユニット 70 から提供される選択情報を用いる。プロセッサ 60 は、プロセッサ 62 とコントローラ 64 とを有する。ユニット 62 は、番組ガイドとシステム情報を含む特定番組情報を処理（すなわち、解析、照合及び組み立て）し、コントローラ 64 は、復号器 100 の動作に必要な残りの制御機能を実行する。ユニット 60 の機能は図 11 に示したように別々の要素 62 と 64 として実装されてもよいが、代替的に単一のプロセッサ内に実装されてもよい。例えば、ユニット 62 と 64 の機能はマイクロプロセッサのプログラム命令に組み込まれてもよい。プロセッサ 60 は、入力信号のフォーマットと符号化タイプを復調し、復号するよう、プロセッサ 13、復調器 15、復号器 17 及び復号システム 100 を設定する。

【0055】

10

20

30

40

50

図11を詳細に検討してオーディオ、ビデオ及び関連データを表す番組を担う信号で変調され、アンテナ10により受信されたキャリアは、デジタル形式に変換され、入力プロセッサ13により処理される。プロセッサ13は、無線周波数(RF)チューナと、中間周波数(IF)ミキサと、入力信号を後段の処理に適した低周波数帯にダウンコンバートする増幅段とを含む。

【0056】

例えば、ビデオ受信器ユーザがリモートコントロールユニット70を用いて視聴するサブチャンネル(SC)を選択すると仮定する。プロセッサ60は、インターフェース65を介してリモートコントロールユニット70から提供される選択情報を用いて、選択されたサブチャンネルSCに対応する物理的チャンネルを受信するように、復号器100要素を適切に設定する。

【0057】

プロセッサ22に提供される出力データは、番組チャンネルコンテンツと、複数のサブチャンネルを通して配信される多くの番組の特定番組情報を含むトランスポート・データストリームの形式である。

【0058】

プロセッサ22は、復号器17により提供される入来パケットのPacket Identifiers(PIDS)を、サブチャンネルSCで送信されているビデオ、オーディオ及びサブピクチャストリームのPID値と一致するか調べる。これらのPID値は、プロセッサ60によりユニット22内の制御レジスタにプリロードされる。プロセッサ22は、サブチャンネルSCで送信される番組を構成するパケットをキャプチャし、それから、ビデオ復号器25及びオーディオ復号器35にそれぞれ出力するMPEG互換のビデオ及びオーディオストリームを構成する。ビデオ及びオーディオストリームは、選択されたサブチャンネルSC番組コンテンツを表す圧縮されたビデオ及びオーディオデータを含む。

プロセッサ22は、MMT-PF及びMMT-TPヘッダ圧縮などの本原理によるヘッダ圧縮が使われているか検出し、パケットがヘッダ圧縮のための参照ヘッダを提供するか検出する。プロセッサ22は、参照ヘッダを記憶し、それを用いて縮小サイズヘッダを復号する。

【0059】

復号器25は、ユニット22から出力されたパケット化されたMPEG互換ビデオデータを復号及び復元し、復元された番組表現画素データをデバイス50に表示のため提供する。同様に、オーディオプロセッサ35は、ユニット22からのパケット化されたオーディオデータを復号し、復号されたオーディオデータであって関連する復元されたビデオデータと同期したものをデバイス55にオーディオ再生のために提供する。

【0060】

図11のシステムの記憶モードでは、ユニット17からの出力データは復号器100により処理され、MPEG互換データストリームを記憶のため提供する。このモードでは、番組は、ユーザによりリモートコントロールユニット70とインターフェース65とを介して、記憶のため選択される。

【0061】

プロセッサ60は、プロセッサ22と共に、選択された番組のパケット化されたコンテンツデータと、関連する特定番組情報を含むコンポジットMPEG互換データストリームを構成する。コンポジットデータストリームはストレージインターフェース95に出力される。ストレージインターフェース95は、コンポジットデータストリームをバッファし、データのギャップとビットレート変動を低減する。バッファされたデータは、媒体105への記憶に適したように、ストレージデバイス90により処理される。ストレージデバイス90は、チャンネル符号化、インターリーブ及びリードソロモン符号化などの既知のエラー符号化手法を用いて、インターフェース95からのバッファされたデータストリームを符号化し、符号化され記憶に適したデータストリームを生成する。ユニット90は、縮約

10

20

30

40

50

された特定番組情報 (condensed program specific information) を組み込んだ、符号化されたデータストリームを、媒体 105 に記憶する。

【0062】

ここで説明した実施形態は、方法またはプロセス、装置、またはソフトウェアプログラム、データストリーム、又は信号として実施できる。1つの形式の実施形態の場合で説明した（例えば、方法としてのみ説明した）場合であっても、説明した機能の実施形態は他の形式（例えば、装置やプログラム）でも実施できる。装置は例えば適切なハードウェア、ソフトウェア、及びファームウェアで実施可能である。上記の方法は、例えばプロセッサ等の装置で実施可能である。プロセッサとは、処理装置一般を指し、例えばコンピュータ、マイクロプロセッサ、集積回路、プログラマブル論理デバイスなどを指す。プロセッサは、エンドユーザ間での情報通信を行う、コンピュータ、セルラー電話、ポータブル／パーソナル・デジタル・アシスタント（PDA）などのデバイス、及びその他の通信デバイスも含む。

【0063】

本原理の「一実施形態」等と言う場合、本発明の少なくとも1つの実施形態に含まれるその実施形態に関して説明する具体的な特徴、構造、特性などを意味する。それゆえ、本明細書を通していろいろなところに記載した「一実施形態において」等と言った場合、必ずしもすべてが同じ実施形態を参照するものではない。

【0064】

また、本願とその特許請求の範囲において、様々な情報を「判断（determining）」する旨を記載した。情報の判断には、例えば、その情報の推定、その情報の計算、その情報の予測、またはその情報のメモリからの読み出しのうちの一以上が含まれ得る。

【0065】

さらに、本願とその特許請求の範囲において、様々な情報を「アクセス（accessing）」する旨を記載した。情報へのアクセスは、例えば、情報の受け取り、（例えば、メモリからの）情報の読み出し、情報の記憶、情報の処理、情報の送信、情報の移動、情報のコピー、情報の削除、情報の計算、情報の決定、情報の予測、又は情報の推定などの一以上を含み得る。

【0066】

また、本願とその特許請求の範囲において、様々な情報を「受け取る」旨を記載した。受け取りは、「アクセス」と同様に、広い意味で用いている。情報の受け取りは、例えば、情報へのアクセス、または（例えば、メモリからの）情報の読み出しのうちの一または複数を含み得る。さらに、「受け取り（receiving）」は、一般的に、例えば、情報の記憶、情報の処理、情報の送信、情報の移動、情報のコピー、情報の削除、情報の計算、情報の決定、情報の予測または情報の推定などの動作中にいろいろな点で関係してくれる。

【0067】

当業者には言うまでもないが、実施形態は、例えば記憶または送信され得る情報を担うようフォーマットされた種々の信号を生成することもできる。情報には、例えば、方法を実行する命令や、説明した実施形態により生成されるデータが含まれ得る。例えば、信号は説明した実施形態のビットストリームを担うようにフォーマットされてもよい。かかる信号は、（例えば、無線周波数のスペクトルを用いた）電磁波やベースバンド信号などとしてフォーマットし得る。フォーマット化には、例えば、データストリームの符号化、符号化したデータストリームによるキャリアの変調が含まれる。信号が担う情報は例えばアナログ情報やデジタル情報であってもよい。知られているように、信号は様々な異なる有線リンクまたは無線リンクで送信できる。信号はプロセッサ読み取り可能媒体に記憶してもよい。

次の付記を記す。

10

20

30

40

50

(付記 1) データを送信する方法であって、

複数のトランSPORTパケットを含むデータストリームに、ビデオデータとオーディオデータのうち少なくとも一方をカプセル化するステップであって、各トランSPORTパケットはヘッダ部とペイロード部とを含むステップと、

フルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのうち一方を用いて、あるトランSPORTパケットを送信できるか判断するステップと、

前記判断するステップに応じて前記データストリームのトランSPORTパケットを構成するステップと、

構成した前記トランSPORTパケットを送信するステップとを含む、方法。

(付記 2) 前記構成するステップは、

第1のヘッダを有する第1のパケットを構成するステップであって、前記第1のヘッダは第2のパケットの第2のヘッダに対する参照として提供され、前記第1のヘッダは前記フルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは前記縮小サイズパケットヘッダフォーマットに対応するステップと、

前記第2のヘッダを生成するステップであって、前記第2のヘッダ中のーフィールドが前記第1のヘッダ中の対応するフィールドに対して表し方が異なるステップとを含む、付記1に記載の方法。

(付記 3) データを受信する方法であって、

複数のトランSPORTパケットを含み、ビデオデータとオーディオデータのうち少なくとも一方を含むデータストリームを受け取るステップであって、各トランSPORTパケットはヘッダ部とペイロード部とを含むステップと、

トランSPORTパケットがフルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのうち一方を用いているか判断するステップと、

判断された前記フォーマットに応じて前記データストリームのトランSPORTパケットを復号するステップとを含む、方法。

(付記 4) 前記受信するステップは、

第1のヘッダを含む第1のパケットを受信するステップと、

第2のヘッダを含む第2のパケットを受信するステップであって、前記第1のヘッダは前記フルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは前記縮小サイズパケットヘッダフォーマットに対応し、前記第2のヘッダはフィールドを含む、ステップとを含み、

前記復号するステップは、

前記第1のヘッダを前記第2のヘッダの参照として用いることを決定するステップと、

前記第2のヘッダ中のフィールドを、前記第1のヘッダ中の対応するフィールドに応じて復号するステップとを含む、

付記3に記載の方法。

(付記 5) データを送信する装置であって、

複数のトランSPORTパケットを含むデータストリームに、ビデオデータとオーディオデータのうち少なくとも一方をカプセル化するように構成され、各トランSPORTパケットはヘッダ部とペイロード部とを含む送信器と、

フルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのうち一方を用いてトランSPORTパケットを送信できるか決定し、前記決定に応じて前記データストリームのトランSPORTパケットを構成するように構成されたヘッダ圧縮器とを有し、前記送信器は構成された前記トランSPORTパケットを送信するように構成される、装置。

(付記 6) 前記送信器は、第1のヘッダを有する第1のパケットを送信するように構成され、前記第1のヘッダは第2のパケットの第2のヘッダに対する参照として提供され、前記第1のヘッダは前記フルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは前記縮小サイズパケットヘッダフォーマットに対応し、

前記ヘッダ圧縮器は前記第2のヘッダを生成するように構成され、前記第2のヘッダ中

10

20

30

40

50

のフィールドは前記第1のヘッダ中の対応するフィールドに対する差として表され、前記送信器は前記第2のヘッダを含む前記第2のパケットを送信する、
付記5に記載の装置。

(付記7) 前記第2のヘッダ中のフィールドは複数のビットで表すことができ、前記ヘッダ圧縮器は、前記第2のヘッダ中のフィールドを前記複数のビットの下位ビットで表すことにより前記第2のヘッダを生成する、
付記2または5に記載の方法または装置。

(付記8) 前記第1のヘッダは、前記フィールドに表されるデータが前記フィールドに含まれるビット数の最大表現能力を超えたときに構成される、
付記2または5に記載の方法または装置。

(付記9) データを受信する装置であって、

複数のトランスポートパケットを含み、ビデオデータとオーディオデータのうち少なくとも一方を含むデータストリームを受け取るように構成され、各トランスポートパケットはヘッダ部とペイロード部とを含む受信器と、

トランスポートパケットがフルサイズパケットヘッダフォーマットと縮小サイズパケットヘッダフォーマットのうち一方を用いているか決定し、決定されたフォーマットに応じて前記データストリームのトランスポートパケットを復号するように構成されたヘッダ復元器とを有する、装置。

(付記10) 前記受信器は第1のヘッダを含む第1のパケットを受信し、第2のヘッダを含む第2のパケットを受信し、前記第1のヘッダは前記フルサイズパケットヘッダフォーマットに対応し、前記第2のヘッダは前記縮小サイズパケットヘッダフォーマットに対応し、前記第2のヘッダはフィールドを含み、前記ヘッダ復元器は、前記第1のヘッダを前記第2のヘッダの参照として用いることを決定し、前記第2のヘッダ中のフィールドを前記第1のヘッダ中の対応するフィールドに応じて復号するように構成される、

付記9に記載の装置。

(付記11) 前記第1のヘッダは前記第1のヘッダが前記参照として用いられることを示す第2のフィールドを含む、

付記2、4、6または10に記載の方法または装置。

(付記12) 前記第1のヘッダ中の対応するフィールドは複数のビットにより表すことができ、前記第2のヘッダ中のフィールドは、前記複数のビットの上位ビットと受信したフィールドとに応じて復号され、前記第2のヘッダ中の復号されたフィールドは、前記第1のヘッダ中の対応するフィールドを表すのに使われるのと同じビット数を有する、
付記4または10に記載の方法または装置。

(付記13) 前記第2のヘッダは前記第1のパケットへのリンクを含む、

付記2、4、6または10に記載の方法または装置。

(付記14) 前記第1のヘッダ中のシーケンス番号は複数のビットにより表すことができ、前記第2のヘッダは前記複数のビットの、前記第1のパケットへのリンクを提供する部分を含む、
付記13に記載の方法または装置。

(付記15) 前記第2のパケット中の第2のヘッダは前記第1のパケット中の第1のヘッダより少ないフィールドを含む、
付記2、4、6または10に記載の方法または装置。

(付記16) 付記1-4、7、8、及び11-15に記載の、データを送信または受信する命令を記憶したコンピュータ読み取り可能記憶媒体。

【図1A】

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
| length | type | if_iF|A|R|P|E|S|+
+-----+-----+-----+-----+-----+-----+-----+
| data_offset | frag_count | numDU | DU_offset |
+-----+-----+-----+-----+-----+-----+-----+
| ... | DU_offset | ... | ...
+-----+-----+-----+-----+-----+-----+
| payload_sequence_number |
+-----+-----+-----+-----+-----+-----+
| header_extension |
+-----+-----+-----+-----+-----+
| payload data |
+-----+-----+-----+-----+-----+

```

FIG. 1A

【図1B】

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
| type | length |
+-----+-----+-----+-----+-----+-----+-----+
| header_extension_value |
+-----+-----+-----+-----+-----+

```

FIG. 1B

【図2】

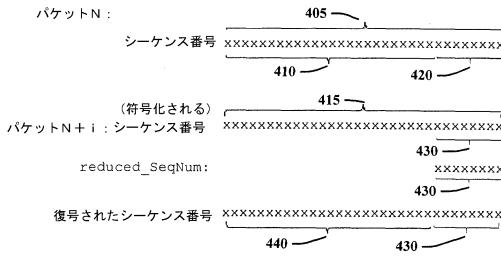
```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
| packet_id | packet_sequence_number |
+-----+-----+-----+-----+-----+-----+-----+
| packet_sequence_number | timestamp |
+-----+-----+-----+-----+-----+-----+-----+
| timestamp | Q|Q|F|P|FEC| RES |TB | DS |R|S|I|
+-----+-----+-----+-----+-----+-----+-----+
| TP | flow_label | reserved | private_user_data |
+-----+-----+-----+-----+-----+-----+-----+
| payload data |
+-----+-----+-----+-----+-----+-----+

```

FIG. 2

【図4】



【図5A】

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
|C|I|Q|Q|F|P|FEC| TB | DS |R|S|I| packet_id |
+-----+-----+-----+-----+-----+-----+-----+
| packet_sequence_number |
+-----+-----+-----+-----+-----+-----+
| timestamp |
+-----+-----+-----+-----+-----+-----+
| TP | flow_label | reserved | private_user_data |
+-----+-----+-----+-----+-----+-----+
| payload data |
+-----+-----+-----+-----+-----+

```

FIG. 5A

【図5B】

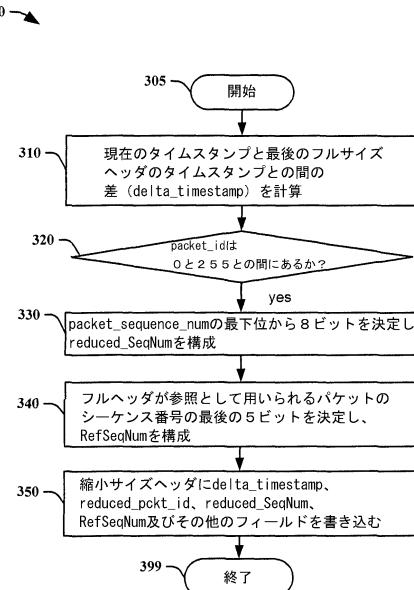
```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
|C|Q|F|P|FEC| TB | DS |R| delta_timestamp |
+-----+-----+-----+-----+-----+-----+-----+
| reduced_packet_id | reduced_SequenceNum | RefSeqNum | TP | flow_label | reserved | ...
+-----+-----+-----+-----+-----+-----+-----+
| private_user_data | payload data | ...
+-----+-----+-----+-----+-----+-----+

```

FIG. 5B

【図3】



【図6A】

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
|C|I|f_i|A|R|P|E| length | type | if_iF|A|R|P|E|S|+
+-----+-----+-----+-----+-----+-----+-----+
| data_offset | frag_count | numDU | DU_offset |
+-----+-----+-----+-----+-----+-----+-----+
| ... | DU_offset | ... | ...
+-----+-----+-----+-----+-----+-----+
| payload_sequence_number |
+-----+-----+-----+-----+-----+-----+
| header_extension |
+-----+-----+-----+-----+-----+
| payload data |
+-----+-----+-----+-----+-----+

```

FIG. 6A

【図6B】

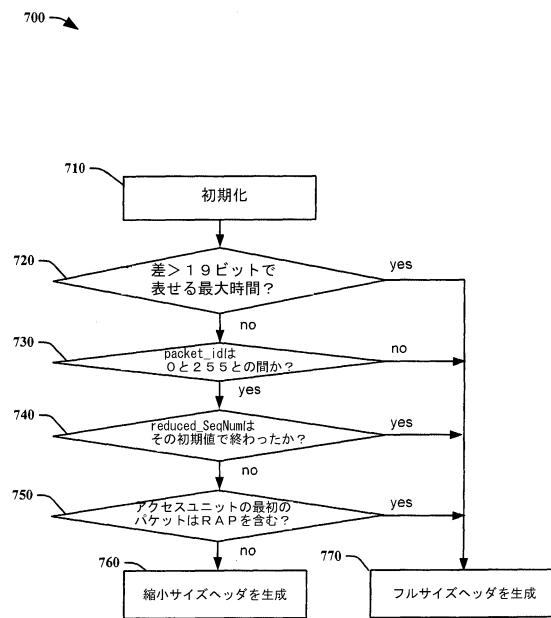
```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+
|C|I|f_i|S| type | data_offset | frag_count | RefSeqNum |
+-----+-----+-----+-----+-----+-----+-----+
| header_extension |
+-----+-----+-----+-----+-----+
| payload data |
+-----+-----+-----+-----+-----+

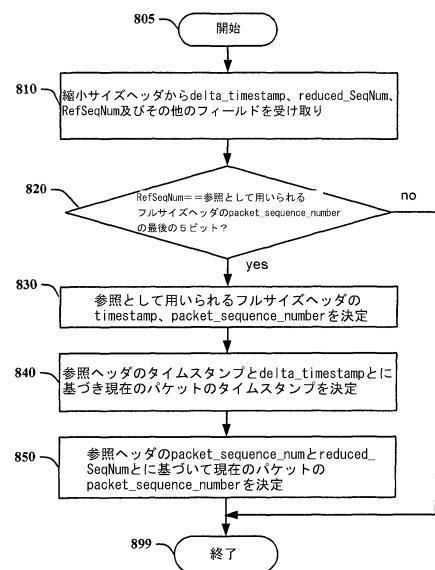
```

FIG. 6B

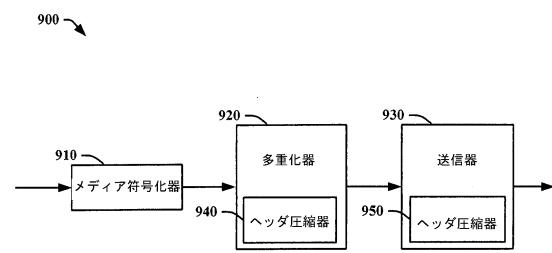
【図7】



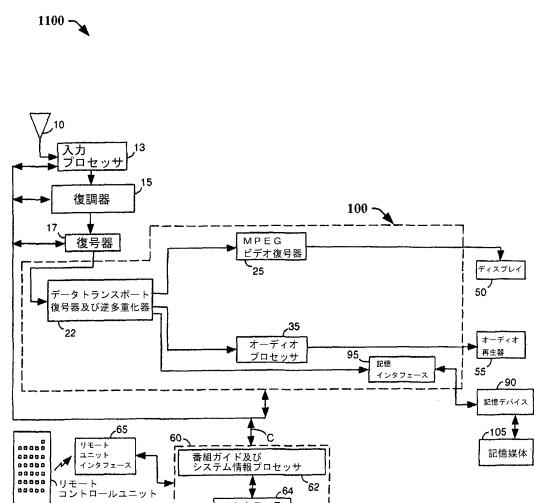
【図8】



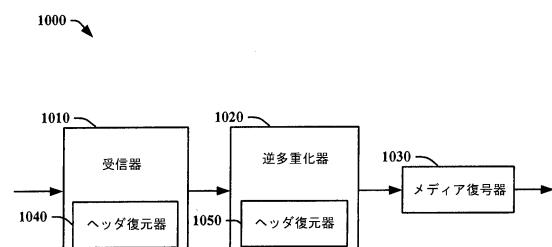
【図9】



【図11】



【図10】



フロントページの続き

(74)代理人 100134120

弁理士 内藤 和彦

(74)代理人 100108213

弁理士 阿部 豊隆

(72)発明者 シャンペル, マリー - リュック

フランス国, エフ - 35220 マルピレ, リュ・ド・ラ・クロワ・ブシェ 4

審査官 松崎 孝大

(56)参考文献 特開2003-008644 (JP, A)

特開2001-320419 (JP, A)

米国特許出願公開第2003/0193950 (US, A1)

特開2003-304299 (JP, A)

吉村 健 Takeshi Yoshimura, 移動通信におけるパケットロスを考慮したRTPヘッダ圧縮・

復元方法 Packet Loss Resilient RTP Header Compression over Wireless Links, マルチメデ

ィア, 分散, 協調とモバイル (DICOMO) シンポジウム論文集 1997年~2006年版

Ver. 1.1 [DVD-ROM] Multimedia, Distributed, Cooperative and Mobile Symposium, 日本, 社団法人情報処理学会, 2000年 6月28日, 第2000巻, pp.295-300