

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2018-201219

(P2018-201219A)

(43) 公開日 平成30年12月20日(2018.12.20)

(51) Int.Cl.
H04N 21/2362 (2011.01)F I
H04N 21/2362テーマコード (参考)
5C164

審査請求 有 請求項の数 1 O L (全 28 頁)

(21) 出願番号 特願2018-145225 (P2018-145225)
 (22) 出願日 平成30年8月1日(2018.8.1)
 (62) 分割の表示 特願2014-196425 (P2014-196425)
 の分割
 原出願日 平成26年9月26日(2014.9.26)
 (31) 優先権主張番号 特願2013-202387 (P2013-202387)
 (32) 優先日 平成25年9月27日(2013.9.27)
 (33) 優先権主張国 日本国(JP)

特許法第30条第2項適用申請有り 第30条第2項適用、平成25年9月6日鳥取大学において開催された一般社団法人情報処理学会第12回情報科学技術フォーラムで発表

(71) 出願人 000004352
 日本放送協会
 東京都渋谷区神南2丁目2番1号
 (74) 代理人 100141139
 弁理士 及川 周
 (74) 代理人 100171446
 弁理士 高田 尚幸
 (74) 代理人 100114937
 弁理士 松本 裕幸
 (74) 代理人 100171930
 弁理士 木下 郁一郎
 (72) 発明者 青木 秀一
 東京都世田谷区砧一丁目10番11号 日本放送協会放送技術研究所内

最終頁に続く

(54) 【発明の名称】 制御メッセージ生成装置、配信システム、受信装置およびそれらのプログラム

(57) 【要約】

【課題】 受信可能になるまでのホップ数の期待値を抑えることができる制御メッセージ生成装置を提供すること。

【解決手段】 第1のパッケージを構成するアセット各々への参照情報を含む第1のテーブルと、第2のパッケージを構成するアセット各々への参照情報を含む第2のテーブルとを生成するテーブル生成部と、第2のテーブルへの参照情報を含むパッケージリストテーブルを生成するパッケージリストテーブル生成部と、第1のテーブルおよびパッケージリストテーブルを含む第1の制御メッセージと、第2のテーブルを含む第2の制御メッセージとを生成する制御メッセージ生成部とを具備することを特徴とする制御メッセージ生成装置である。

【選択図】 図1

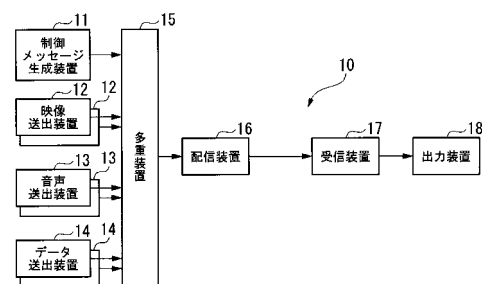


図1

【特許請求の範囲】

【請求項 1】

第 1 のパッケージを構成するアセット各々への参照情報を含む第 1 のテーブルと、第 2 のパッケージを構成するアセット各々への参照情報を含む第 2 のテーブルとを生成するテーブル生成部と、

前記第 2 のテーブルへの参照情報を含むパッケージリストテーブルを生成するパッケージリストテーブル生成部と、

前記第 1 のテーブルおよび前記パッケージリストテーブルを含む第 1 の制御メッセージと、前記第 2 のテーブルを含む第 2 の制御メッセージとを生成する制御メッセージ生成部と、

10

を具備し、

前記第 2 のテーブルへの参照情報は、

前記第 2 のパッケージを示すパッケージ ID と、

前記第 2 のテーブルを含む前記第 2 の制御メッセージを格納するパケットのパケット ID と、

であることを特徴とする制御メッセージ生成装置。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、制御メッセージ生成装置、配信システム、受信装置およびそれらのプログラムに関する。

20

【背景技術】

【0002】

地上デジタルテレビジョン放送など、現行のデジタルテレビジョン放送では、ISO/IEC 13818-1 として規格化されている MPEG (Moving Picture Experts Group) 2-TS (Transport Stream) が用いられている。これらのデジタルテレビジョン放送では、映像や音声のストリームに加えて、PAT (Program Association Table)、PMT (Program Map Table) と呼ばれるテーブルが、それぞれ周期的に伝送されている (例えば、非特許文献 1、2 参照)。

【0003】

30

PMT は、番組毎に用意されるテーブルであり、番組を構成する映像ストリームや、音声ストリームへの参照情報である PID が格納されている。PAT は、MPEG 2-TS 内に一つだけ用意されるテーブルであり、その MPEG 2-TS により伝送されている PMT 各々への参照情報である PID が格納されている。

【0004】

デジタルテレビジョン放送を受信する受信装置は、ある番組を受信する際には、まず、MPEG 2-TS から PAT を取得する。この PAT を参照すると、受信する番組の PMT の PID が分かるので、受信装置は、受信する番組の PMT を、MPEG 2-TS から取得する。そして、この PMT を参照すると、受信する番組の映像や音声のストリームの PID が分かるので、受信装置は、受信する番組の映像や音声のストリームを、MPEG 2-TS から取得し、映像の表示や音声の出力を行う。

40

【先行技術文献】

【非特許文献】

【0005】

【非特許文献 1】“デジタル放送における映像符号化、音声符号化及び多重化方式”、ARRIB STD-B32 2. 8 版、社団法人 電波産業会、平成 24 年 12 月 18 日

【非特許文献 2】“デジタル放送に使用する番組配列情報”、ARRIB STD-B10 4. 8 版、社団法人 電波産業会、平成 22 年 4 月 26 日

【発明の概要】

【発明が解決しようとする課題】

50

【0006】

しかしながら、上述のデジタルテレビジョン放送など、MPEG2-TSを用いたシステムにおいては、上述したように、受信装置が、番組を構成する映像や音声のストリームを受信可能になるまでには、PATと、PMTとを取得するという2ホップが常に必要であるという問題がある。2ホップ必要であるため、受信する番組が指定されてから、映像や音声のストリームを受信するまでにかかる時間の期待値は、PATが送信されている周期と、PMTが送信されている周期との和となる。したがって、該時間を短くしようとすると、PATおよびPMTの送信周期を短くすることになり、映像や音声の帯域を圧迫してしまう。また、映像や音声の帯域への圧迫を減らそうとすると、PATまたはPMTの送信周期を長くすることになり、受信する番組が指定されてから、映像や音声のストリームを受信するまでにかかる時間が長くなってしまう。

10

【0007】

本発明は、このような事情に鑑みてなされたもので、受信可能になるまでのホップ数の期待値を抑えることができる制御メッセージ生成装置、配信システム、受信装置およびプログラムを提供する。

【課題を解決するための手段】

【0008】

(1) この発明は上述した課題を解決するためになされたもので、本発明の一対応は、第1のパッケージを構成するアセット各々への参照情報を含む第1のテーブルと、第2のパッケージを構成するアセット各々への参照情報を含む第2のテーブルとを生成するテーブル生成部と、前記第2のテーブルへの参照情報を含むパッケージリストテーブルを生成するパッケージリストテーブル生成部と、前記第1のテーブルおよび前記パッケージリストテーブルを含む第1の制御メッセージと、前記第2のテーブルを含む第2の制御メッセージとを生成する制御メッセージ生成部とを具備することを特徴とする制御メッセージ生成装置である。

20

【0009】

(2) また、本発明の他の態様は、(1)に記載の制御メッセージ生成装置であって、前記第2のテーブルへの参照情報は、前記第2のテーブルを含む前記第2の制御メッセージを格納するパケットのパケットIDであることを特徴とする。

【0010】

30

(3) また、本発明の他の態様は、第1のパッケージを構成するアセット各々への参照情報を含む第1のテーブルと、第2のパッケージを構成するアセット各々への参照情報を含む第2のテーブルへの参照情報を含むパッケージリストテーブルとを含む第1の制御メッセージと、前記第2のテーブルを含む第2の制御メッセージとを生成する制御メッセージ生成装置と、前記第1の制御メッセージと、前記第2の制御メッセージとを、データフローを用いて配信する配信装置とを具備することを特徴とする配信システムである。

【0011】

(4) また、本発明の他の態様は、第1のパッケージを構成するアセット各々への参照情報を含む第1のテーブルと第2のパッケージを構成するアセット各々への参照情報を含む第2のテーブルへの参照情報を含むパッケージリストテーブルとを含む第1の制御メッセージと、前記第2のテーブルを含む第2の制御メッセージとを含むデータフローを受信する受信装置であって、所望のパッケージを指定する情報を取得する指定取得部と、前記データフローを受信する受信部と、前記データフローから前記第1の制御メッセージを分離する分離部と、前記第1の制御メッセージに含まれる前記第1のテーブルを参照して、前記第1のパッケージが前記所望のパッケージと一致しているときは、前記第1のパッケージの分離を前記分離部に指示する制御メッセージ処理部とを具備することを特徴とする受信装置である。

40

【0012】

(5) また、本発明の他の態様は、(4)に記載の受信装置であって、前記制御メッセージ処理部は、前記第1の制御メッセージに含まれる前記第1のテーブルを参照して、前記

50

第 1 のパッケージが前記所望のパッケージと一致していないときは、前記第 1 の制御メッセージに含まれる前記パッケージリストテーブルを参照して、前記所望のパッケージに対応する前記第 2 のテーブルへの参照情報を取得し、取得した参照情報を用いて、前記所望のパッケージに対応する前記第 2 のテーブルを含む前記第 2 の制御メッセージの分離を前記分離部に指示することを特徴とする。

【0013】

(6) また、本発明の他の態様は、コンピュータを、第 1 のパッケージを構成するアセット各々への参照情報を含む第 1 のテーブルと、第 2 のパッケージを構成するアセット各々への参照情報を含む第 2 のテーブルとを生成するテーブル生成部、前記第 2 のテーブルへの参照情報を含むパッケージリストテーブルを生成するパッケージリストテーブル生成部、前記第 1 のテーブルおよび前記パッケージリストテーブルを含む第 1 の制御メッセージと、前記第 2 のテーブルを含む第 2 の制御メッセージとを生成する制御メッセージ生成部として機能させるためのプログラムである。

10

【0014】

(7) また、本発明の他の態様は、第 1 のパッケージを構成するアセット各々への参照情報を含む第 1 のテーブルと第 2 のパッケージを構成するアセット各々への参照情報を含む第 2 のテーブルへの参照情報を含むパッケージリストテーブルとを含む第 1 の制御メッセージと、前記第 2 のテーブルを含む第 2 の制御メッセージとを含むデータフローを受信する受信装置のコンピュータを、所望のパッケージを指定する情報を取得する指定取得部、前記データフローを受信する受信部、前記データフローから前記第 1 の制御メッセージを分離する分離部、前記第 1 の制御メッセージに含まれる前記第 1 のテーブルを参照して、前記第 1 のパッケージが前記所望のパッケージと一致しているときは、前記第 1 のパッケージの分離を前記分離部に指示する制御メッセージ処理部として機能させるためのプログラムである。

20

【発明の効果】

【0015】

この発明によれば、受信可能になるまでのホップ数の期待値を抑えることができる。

【図面の簡単な説明】

【0016】

【図 1】この発明の一実施形態による配信システム 10 の構成を示す概略ブロック図である。

30

【図 2】同実施形態における制御メッセージ生成装置 11 の構成を示す概略ブロック図である。

【図 3】同実施形態における受信装置 17 の構成を示す概略ブロック図である。

【図 4】同実施形態における配信装置 16 が配信する IP データフローの構成を示す概念図である。

【図 5】同実施形態における PA メッセージのデータ構造を示す表である。

【図 6】同実施形態における MP テーブルのデータ構造を示す表（その 1）である。

【図 7】同実施形態における MP テーブルのデータ構造を示す表（その 2）である。

【図 8】同実施形態における MP テーブルのデータ構造を示す表（その 3）である。

40

【図 9】同実施形態における参照情報のデータ構造を示す表（その 1）である。

【図 10】同実施形態における参照情報のデータ構造を示す表（その 2）である。

【図 11】同実施形態における参照情報のデータ構造を示す表（その 3）である。

【図 12】同実施形態におけるパッケージリストテーブルのデータ構造を示す表（その 1）である。

【図 13】同実施形態におけるパッケージリストテーブルのデータ構造を示す表（その 2）である。

【図 14】同実施形態における指定取得部 72、受信制御部 73、制御メッセージ処理部 74 の動作を説明するフローチャートである。

【図 15】同実施形態の変形例 1 における配信装置 16 が配信する IP データフローの構

50

成を示す概念図である。

【図 1 6】同実施形態の変形例 2 における M P テーブルのデータ構造を示す表（その 1）である。

【図 1 7】同実施形態の変形例 2 における M P テーブルのデータ構造を示す表（その 2）である。

【図 1 8】同実施形態の変形例 2 における M P テーブルのデータ構造を示す表（その 3）である。

【図 1 9】同実施形態の変形例 2 における参照情報のデータ構造を示す表（その 1）である。

【図 2 0】同実施形態の変形例 2 における参照情報のデータ構造を示す表（その 2）である。 10

【図 2 1】同実施形態の変形例 2 におけるパッケージリストテーブルのデータ構造を示す表（その 1）である。

【図 2 2】同実施形態の変形例 2 におけるパッケージリストテーブルのデータ構造を示す表（その 2）である。

【発明を実施するための形態】

【0 0 1 7】

以下、図面を参照して、本発明の実施の形態について説明する。図 1 は、この発明の一実施形態による配信システム 1 0 の構成を示す概略ブロック図である。配信システム 1 0 は、制御メッセージ生成装置 1 1、複数の映像送出装置 1 2、複数の音声送出装置 1 3、 20 複数のデータ送出装置 1 4、多重装置 1 5、配信装置 1 6、受信装置 1 7、出力装置 1 8 を含んで構成される。配信システム 1 0 は、I P データフローを用いて、受信装置 1 7 にコンテンツを配信する。なお、I P データフローは、例えば、U D P / I P を用いたマルチキャストのフローであり、送信元の I P アドレスと、宛先の I P アドレスと、宛先のポート番号とで決まる。また、I P データフローは、インターネットなどの双方向の伝送路により伝送されてもよいし、地上デジタル放送や衛星デジタル放送などの片方向の伝送路により伝送されてもよい。

【0 0 1 8】

制御メッセージ生成装置 1 1 は、M M T (MPEG Media Transport) の制御メッセージ (Signaling Message) を生成し、多重装置 1 5 に入力する。本実施形態において、制御メ 30 ッセージ生成装置 1 1 が生成する制御メッセージは、M P テーブル (MMT Package table) と、パッケージリストテーブルを含む P A メッセージ (Package Access message) と、M P テーブルのみを含む P A メッセージである。M P テーブルは、対象のパッケージのパッケージ I D と、該パッケージを構成するアセットへの参照情報 (MMT_general_location_info) を格納している。パッケージリストテーブルは、当該パッケージリストテーブルを伝送しているデータフローに含まれるパッケージ各々の M P テーブルへの参照情報 (パケット I D など) を格納している。本実施形態では、パッケージリストテーブルは、当該パッケージリストテーブルとともに P A メッセージに含まれている M P テーブルへの参照情報を含んでいないが、含んでいてもよい。

【0 0 1 9】 40

P A メッセージは、M M T における制御メッセージ (Signaling Message) の一種である。アセットとは、映像ストリーム、音声ストリーム、データストリーム、ハイパーテキストファイル、画像ファイルなどのメディアである。パッケージは、コンテンツを構成する 1 つまたは複数のアセットの集合である。例えば、ある映画のパッケージを構成するアセットは、映像ストリームと、音声ストリームと、字幕のデータストリームである。なお、映像ストリームに、モバイル用の低解像度のものと、据え置き型用の高解像度のものがあるときには、パッケージには、両方が含まれる。なお、パッケージを構成するアセットの一部は、配信装置 1 6 とは異なる装置から、配信装置 1 6 と受信装置 1 7 との間の伝送路とは異なる伝送路を用いて配信されてもよい。なお、M P テーブル、パッケージリストテーブル、P A メッセージの詳細は、後述する。 50

【0020】

映像送出装置12は、コンテンツを構成するアセットのうち、映像のストリームを生成し、多重装置15に入力する。音声送出装置13は、コンテンツを構成するアセットのうち、音声のストリームを生成し、多重装置15に入力する。データ送出装置14は、コンテンツを構成するアセットの内、データのストリームを生成し、多重装置15に入力する。データのストリームは、例えば、映像に付随して表示される情報や、映像に重畳する字幕などである。

【0021】

多重装置15は、入力された制御メッセージ、映像のストリーム、音声のストリーム、データのストリームを、それぞれMMTパケットに格納し、該MMTパケットを時系列に並べて多重化する。多重装置15は、MMTパケットのパケットIDを、格納しているストリームや制御メッセージ毎に予め設定された値とする。特に、多重装置15は、MMTパケットのうち、MPテーブルとパッケージリストテーブルとの両方を格納しているPAMESSAGEを格納したもののパケットIDを「0」とする。配信装置16は、多重装置15が多重化したMMTパケットを、IPパケットに格納してIPデータフローを生成し、配信する。

【0022】

受信装置17は、配信装置16が配信したIPデータフローを受信する。受信装置17は、受信したIPデータフローに含まれるコンテンツの復号信号を生成し、出力装置18に入力する。出力装置18は、例えば、液晶ディスプレイと、スピーカとを備え、入力された復号信号が表す映像を液晶ディスプレイに表示し、該復号信号が表す音声をスピーカから出力する。

【0023】

図2は、制御メッセージ生成装置11の構成を示す概略ブロック図である。制御メッセージ生成装置11は、MPテーブル生成部21、パッケージリストテーブル生成部22、PAMESSAGE生成部23を含んで構成される。MPテーブル生成部21は、配信装置16が送信するパッケージ各々のMPテーブルを生成する。パッケージリストテーブル生成部22は、MPテーブル生成部21が生成するMPテーブル各々に対応するパッケージIDと、該MPテーブルを格納するMMTパケットのパケットIDとを含むパッケージリストテーブルを生成する。なお、パッケージリストテーブルは、当該パッケージリストテーブルと同じPAMESSAGEに格納するMPテーブルのパッケージIDと、パケットIDとを含んでいなくてもよい。

【0024】

PAMESSAGE生成部23（制御メッセージ生成部）は、MPテーブル生成部21が生成したMPテーブルを格納したPAMESSAGEを生成する。ただし、特定のMPテーブルを格納したPAMESSAGEには、PAMESSAGE生成部23は、パッケージリストテーブル生成部22が生成したパッケージリストテーブルも格納する。PAMESSAGE生成部23は、生成したPAMESSAGEを多重装置15に入力する。

【0025】

図3は、受信装置17の構成を示す概略ブロック図である。受信装置17は、受信部70、分離部71、指定取得部72、受信制御部73、制御メッセージ処理部74、映像デコード部75、音声デコード部76、データレンダリング部77、映像合成部78、音声合成部79を含んで構成される。

【0026】

受信部70は、受信制御決定部73からの指示に従い、配信装置16により配信されたIPデータフローを受信する。分離部71は、受信部70が受信したIPデータフローから、パケットIDが「0」のMMTパケットに含まれるPAMESSAGEを分離し、制御メッセージ処理部74に入力する。また、分離部71は、受信部70が受信したIPデータフローから、制御メッセージ処理部74に指定された制御メッセージと、映像ストリームと、音声ストリームと、データストリームとを分離する。分離部71は、分離した制御メ

10

20

30

40

50

ッセージを、制御メッセージ処理部 74 に入力する。分離部 71 は、分離した映像ストリームを、映像デコード部 75 に入力し、分離した音声ストリームを、音声デコード部 76 に入力する。分離部 71 は、分離したデータストリームを、データレンダリング部 77 に入力する。

【0027】

指定取得部 72 は、リモートコントローラなどの入力デバイスを用いてユーザが入力した、受信するサービスの指定を取得する。サービスの指定は、例えば、サービス ID (サービス識別子) をユーザが入力することで行われる。受信制御部 73 は、指定取得部 72 が取得したサービスを伝送している IP データフローの受信を受信部 70 に指示する。また、受信制御部 73 は、指定取得部 72 が取得したサービスのパッケージ ID を、制御メ
10

【0028】

制御メッセージ処理部 74 は、分離部 71 が分離した制御メッセージから、受信制御部 73 から通知されたパッケージ ID の MP テーブルを取得する。制御メッセージ処理部 74 は、取得した MP テーブルを参照して、再生する映像ストリーム、音声ストリーム、データストリームを決定し、決定したストリームを分離部 71 に指定する。受信制御部 73 から通知されたパッケージ ID の MP テーブル取得方法の詳細は後述する。

【0029】

映像デコード部 75 は、分離部 71 から入力された映像ストリームを復号し、映像信号
20

を生成する。音声デコード部 76 は、分離部 71 から入力された音声ストリームを復号し、音声信号を生成する。データレンダリング部 77 は、分離部 71 から入力されたデータストリームをレンダリングして、映像信号と、音声信号とを生成する。また、データレンダリング部 77 は、生成した映像信号と、映像デコード部 75 が生成した映像信号との合成方法を指定する信号も、データストリームをレンダリングすることで生成する。

【0030】

例えば、データストリームが字幕のデータストリームであれば、データレンダリング部 77 が生成する信号が指定する合成方法は、映像デコード部 75 が生成した映像信号が表
30

す映像に、データレンダリング部 77 が生成した映像信号が表す字幕を重畳するという方法となる。また、データストリームが、映像デコード部 75 が生成した映像信号が表す映像の左外側と、下外側とに情報を表示するものであれば、データレンダリング部 77 が生成する信号が指定する合成方法は、映像デコード部 75 が生成した映像信号が表す映像を縮小し、該縮小した映像の左外側と下外側とに、データレンダリング部 77 が生成した映像信号が表す映像を配置するという方法となる。

【0031】

映像合成部 78 は、映像デコード部 75 が生成した映像信号と、データレンダリング部 77 が生成した映像信号とを、データレンダリング部 77 が生成した合成方法を指定する
40

信号に従い合成して、出力装置 18 に表示させる映像の映像信号を生成する。音声合成部 79 は、音声デコード部 76 が生成した音声信号と、データレンダリング部 77 が生成した音声信号とを重畳して、出力装置 18 に出力させる音声の音声信号を生成する。

【0032】

図 4 は、配信装置 16 が配信する IP データフローの構成を示す概念図である。図 4 に示す例は、複数のパッケージが一つの IP データフロー DF により伝送されている。IP
データフロー DF は、例えば、伝送媒体が放送波であれば TLV (Type Length Value) ストリームにより伝送され、伝送媒体が LAN (Local Area Network) であれば IP パケッ
ットによって伝送される。

【0033】

IP データフロー DF には、PA メッセージ PA1、PA2、映像ストリーム V1、V2、音声ストリーム A1、A2、データストリーム D1、D2 が含まれる。これらのうち、映像ストリーム V1、音声ストリーム A1、データストリーム D1 で、1 つ目のパッケ
50

ージを構成している。また、映像ストリームV2、音声ストリームA2、データストリームD2で、2つ目のパッケージを構成している。

【0034】

PAメッセージPA1は、1つ目のパッケージのMPテーブルMP1と、パッケージリストテーブルPL1とを格納している。MPテーブルMP1は、1つ目のパッケージのパッケージIDと、該パッケージを構成するアセットである映像ストリームV1、音声ストリームA1、データストリームD1の各々を送送するMMTパケットのパケットIDを格納している。

【0035】

パッケージリストテーブルPL1は、IPデータフローDFにて送送されているパッケージのMPテーブル（例えば、MPテーブルMP2）を格納しているPAメッセージ（例えば、PAメッセージPA2）各々を送送するMMTパケットのパケットIDを格納している。そして、2つ目以降のパッケージのMPテーブル（例えば、MPテーブルMP2）各々は、該パッケージのパッケージIDと、該パッケージを構成するアセット（例えば、映像ストリームV2、音声ストリームA2、データストリームD2）各々を送送するMMTパケットのパケットIDを格納している。

【0036】

なお、PAメッセージPA1は、予め決められたパケットID（例えば、「0」）のMMTパケットに格納されているので、分離部71は、参照情報なしに、IPデータフローDFから分離することができる。IPデータフローDFは、このような構成となっているので、受信装置17は、1つ目のパッケージを受信する際には、パケットIDが「0」のMMTパケットからPAメッセージPA1を抽出し、さらに該PAメッセージPA1からMPテーブルMP1を抽出することで、1つ目のパッケージの構成を把握し、受信することができる。

【0037】

また、2つ目のパッケージを受信する際には、受信装置17は、パケットIDが「0」のMMTパケットからPAメッセージPA1を抽出し、さらに該PAメッセージPA1からパッケージリストテーブルPL1を抽出することで、該パッケージのMPテーブルMP2を格納するMMTパケットのパケットIDを得る。受信装置17は、このパケットIDのMMTパケットからPAメッセージPA2を抽出し、さらに該PAメッセージPA2からMPテーブルMP2を抽出することで、2つ目のパッケージの構成を把握し、受信することができる。3つ目以降のパッケージを受信する際にも、受信装置17は、2つ目のときと同様にすることで、該パッケージの構成を把握し、受信することができる。

【0038】

図5は、PAメッセージのデータ構造を示す表である。本実施形態におけるPAメッセージのデータ構造は、ISO/IEC 23008-1に従っている。該表において、No. は、該表における行の通し番号を示す欄である。データ構造は、PAメッセージのデータ構造を示す欄である。ビット数は、該データ構造のビット数を示す欄である。ビット列表記は、該ビット数のデータ構造のデータ形式を示す欄である。ビット列表記のuim s b fは、unsigned integer most significant bit firstの略であり、符号無し整数、最上位ビットが先頭を意味する。

【0039】

図5において、No. 1の「PA__message() {」は、以降、No. 18の「}」までが、PAメッセージのデータ構造であることを表す。当該メッセージの先頭に配置されているNo. 2の「message__id」は、PAメッセージのID (Identifier) を示す16ビットの領域である。次に配置されているNo. 3の「version」は、PAメッセージのバージョン番号を示す8ビットの領域である。次に配置されているNo. 4の「length」は、当該領域以降のPAメッセージのバイト数を示す32ビットの領域である。次のNo. 5の「extension {」は、以降、No. 12までが制御メッセージに共通のデータ構造に対する、PAメッセージ特有の拡張部分であるこ

とを示す。次に配置されているNo. 6の「number_of_tables」は、当該PAメッセージに格納されているテーブルの数Nを示す8ビットの領域である。

【0040】

次のNo. 7の「for (i = 0 ; i < N ; i++) {」は、以降、「}」であるNo. 11までを、N回、すなわち格納されているテーブルの数分繰り返すことを示す。次に配置されているNo. 8の「table_id」は、当該PAメッセージに格納されているテーブルのIDを示す8ビットの領域である。

【0041】

次に配置されているNo. 9の「table_version」は、直前のNo. 8のテーブルIDが示すテーブルのバージョン番号を示す8ビットの領域である。次に配置されているNo. 10の「table_length」は、直前のNo. 8のテーブルIDが示すテーブルの長さを示す8ビットの領域である。次のNo. 11の「}」は、No. 7の「{」に対応し、No. 12の「}」は、No. 5の「{」に対応する。

【0042】

次のNo. 13の「message_payload {」は、以降、No. 17までに、当該PAメッセージのペイロードが格納されることを示す。次のNo. 14の「for (i = 0 ; i < N1 ; i++) {」は、以降、「}」であるNo. 16までを、N回、すなわち格納されているテーブルの数分繰り返すことを示す。次のNo. 15の「table ()」は、テーブルを配置する領域である。なお、テーブルの配置順は、No. 8の「table_id」の配置順と同じである。次のNo. 16の「}」は、No. 14の「{」に対応し、No. 17の「}」は、No. 13の「{」に対応し、No. 18の「}」は、No. 1の「{」に対応する。

【0043】

図6から図8は、MPテーブルのデータ構造を示す表である。本実施形態におけるMPテーブルのデータ構造は、ISO/IEC 23008-1に従っている。該表において、No. は、該表における行の通し番号を示す欄である。データ構造は、MPテーブルのデータ構造を示す欄である。ビット数は、該データ構造のビット数を示す欄である。値は、該ビット数のデータ構造がとる値を示す欄である。ビット列表記は、該ビット数のデータ構造のデータ形式を示す欄である。ビット列表記のuimsbfは、unsigned integer most significant bit firstの略であり、符号無し整数、最上位ビットが先頭を意味する。また、ビット列表記のbslbfは、bit string left bit firstの略であり、ビット列、左ビットが先頭を意味する。

【0044】

図6において、No. 1の「MPT () {」は、以降、No. 60の「}」までが、MPテーブルのデータ構造であることを表す。当該テーブルの先頭に配置されているNo. 2の「table_id」は、MPテーブルのIDを示す8ビットの領域である。本実施形態では、MPテーブルのIDとして、Complete MPTを表す「0x40」を用いる。次に配置されているNo. 3の「version」は、MPテーブルのバージョン番号を示す8ビットの領域である。次に配置されているNo. 4の「length」は、当該領域以降のMPテーブルのバイト数を示す16ビットの領域である。

【0045】

次に配置されているNo. 5の「reserved」は、6ビットの予約領域である。次に配置されているNo. 6の「MPT_mode」は、MPテーブルがサブセットであるときの処理モードを示す2ビットの領域である。ISO/IEC 23008-1の規定に従うが、詳細な説明は省略する。次のNo. 7の「If (table_id == SUBSET_0_MPT_TABLE_ID) {」は、No. 2の「table_id」の値が、「SUBSET_0_MPT_TABLE_ID」であるときに、No. 20までの領域が配置されることを示す。

【0046】

次のNo. 8の「MMT_package_id {」は、以降、No. 13の「}」ま

10

20

30

40

50

でがパッケージIDに関する領域であることを示す。次に配置されているNo. 9の「MMT_package_id_length」は、パッケージIDのバイト数N1を示す8ビットの領域である。次のNo. 10の「for (i=0; i<N1; i++) {」は、以降、No. 12までの領域をN1回、すなわちパッケージIDのバイト数分繰り返すことを示す。次に配置されているNo. 11の「MMT_package_id_byte」は、パッケージID中の1バイトを示す領域である。次のNo. 12の「}」は、No. 10の「{」に対応し、No. 13の「}」は、No. 8の「{」に対応する。

【0047】

次のNo. 14の「MPT_descriptors {」は、以降、No. 19の「}」までがMPテーブルに関する記述子(descriptor)に関する領域であることを示す。次に配置されているNo. 15の「MPT_descriptors_length」は、配置されている記述子の長さの合計N2を示す16ビットの領域である。次のNo. 16の「for (i=0; i<N2; i++) {」は、以降、No. 18までの領域をN2回、すなわち記述子の長さの合計分繰り返すことを示す。次に配置されているNo. 17の「MPT_descriptors_byte」は、記述子中の1バイトを示す領域である。次のNo. 18の「}」は、No. 16の「{」に対応し、No. 19の「}」は、No. 14の「{」に対応し、No. 20の「}」は、No. 7の「{」に対応する。

【0048】

次に配置されている図7におけるNo. 21の「number_of_assets」は、当該MPテーブルで情報が提供されているアセットの数を示す8ビットの領域である。次のNo. 22の「for (i=0; i<N3; i++) {」は、以降、図8におけるNo. 59までの領域をN3回、すなわち当該MPテーブルで情報が提供されているアセットの数分繰り返すことを示す。

【0049】

次のNo. 23の「asset {」は、以降、No. 29の「}」までがアセットIDに関する領域であることを示す。次に配置されているNo. 24の「asset_id_length」は、配置されているアセットIDのバイト数N4を示す8ビットの領域である。次のNo. 25の「for (i=0; i<N4; i++) {」は、以降、No. 27までの領域をN4回、すなわちアセットIDのバイト数分繰り返すことを示す。次に配置されているNo. 26の「asset_id_byte」は、アセットID中の1バイトを示す領域である。次のNo. 27の「}」は、No. 25の「{」に対応する。

【0050】

次に配置されているNo. 28の「mime_type」は、アセットIDが示されたアセットのMIME (Multipurpose Internet Mail Extension) タイプを示す32ビットの領域である。次のNo. 29の「}」は、No. 23の「{」に対応する。次のNo. 30の「If (asset_scheme_code == "GURL") {」は、「asset_scheme_code」の値が「GURL」であるときに、No. 36までの領域が配置されることを示す。

【0051】

次に配置されているNo. 31の「packet_id」は、GFD (General File Delivery) セッションのIDを示す32ビットの領域である。次に配置されているNo. 32の「number_of_codepoints」は、当該GFDセッションのコードポイント (codepoint) の数N8を示す8ビットの領域である。次のNo. 33の「for (i=0; i<N8; i++) {」は、以降、No. 35までの領域をN8回、すなわちコードポイントの数分繰り返すことを示す。次に配置されているNo. 34の「CodePoint ()」は、コードポイントテーブルを示す領域である。次のNo. 35の「}」は、No. 33の「{」に対応し、No. 36の「}」は、No. 30の「{」に対応する。

【0052】

次に配置されているNo. 37の「reserved」は、7ビットの予約領域である。次に配置されているNo. 38の「asset_clock_relation_flag」は、当該アセットがクロックの参照先として、NTP (Network Time Protocol) を用いるか否かを示す1ビットの領域である。次のNo. 39の「if (asset_clock_relation_flag == 1) {」は、No. 38の「asset_clock_relation_flag」の値が「1」であるときに、No. 46までの領域が配置されることを示す。

【0053】

次に配置されているNo. 40の「asset_clock_relation_id」は、当該アセットが参照するクロックのIDを示す8ビットの領域である。次に配置されている図8におけるNo. 41の「reserved」は、7ビットの予約領域である。次に配置されているNo. 42の「asset_timescale_flag」は、「asset_timescale」が配置されているか否かを示すフラグである。

【0054】

次のNo. 43の「if (asset_timescale_flag == 1) {」は、No. 42の「asset_timescale_flag」の値が「1」であるときに、No. 45までの領域が配置されることを示す。次に配置されているNo. 44の「asset_timescale」は、当該アセットが参照するクロックの周波数を示す32ビットの領域である。次のNo. 45の「}」は、No. 43の「{」に対応し、No. 46の「}」は、No. 39の「{」に対応する。

【0055】

次のNo. 47の「asset_location {」は、以降、No. 52の「}」までが、当該アセットの参照情報（ロケーション情報ともいう）に関する領域であることを示す。次に配置されているNo. 48の「location_count」は、当該アセットに関する参照情報の数N6を示す8ビットの領域である。次のNo. 49の「for (i = 0; i < N6; i++) {」は、以降、No. 51までの領域をN6回、すなわち参照情報の数分繰り返すことを示す。次に配置されているNo. 50の「MMT_general_location_info ()」は、当該アセットに関する参照情報（ロケーション情報ともいう）である。詳細は、後述する。次のNo. 51の「}」は、No. 49の「{」に対応し、No. 52の「}」は、No. 47の「{」に対応する。

【0056】

次のNo. 53の「asset_descriptors {」は、以降、No. 58の「}」までが、当該アセットに関する記述子の領域であることを示す。次に配置されているNo. 54の「asset_descriptors_length」は、当該アセットに関する記述子の長さの合計N5を示す16ビットの領域である。次のNo. 55の「for (i = 0; i < N5; i++) {」は、以降、No. 57までの領域をN5回、すなわち記述子の長さの合計分繰り返すことを示す。次に配置されているNo. 56の「asset_descriptors_byte」は、記述子中の1バイトを示す領域である。次のNo. 57の「}」は、No. 55の「{」に対応し、No. 58の「}」は、No. 53の「{」に対応し、No. 59の「}」は、No. 22の「{」に対応し、No. 60の「}」は、No. 1の「{」に対応する。

【0057】

図9から図11は、参照情報のデータ構造を示す表である。本実施形態における参照情報（general_location_info）のデータ構造は、ISO/IEC 23008-1に従っている。該表において、No. は、該表における行の通し番号を示す欄である。データ構造は、参照情報のデータ構造を示す欄である。ビット数は、該データ構造のビット数を示す欄である。値は、該ビット数のデータ構造がとる値を示す欄である。ビット列表記は、該ビット数のデータ構造のデータ形式を示す欄である。ビット列表記のuimsbfは、unsigned integer most significant bit firstの略であり、符号無し整数、最上位ビットが先頭を意味する。また、ビット列表記のbslbfは、bit st

10

20

30

40

50

ring left bit firstの略であり、ビット列、左ビットが先頭を意味する。また、ビット列表記のcharは、文字を意味する。

【0058】

図9において、No. 1の「MMT_general_location_info (」{」は、以降、図11のNo. 63の「}」までが、参照情報のデータ構造であることを表す。次に配置されているNo. 2の「location_type」は、参照情報のタイプを示す8ビットの領域である。なお、「location_type」が「0x00」であるときは、ロケーション（データパス）が、この「location_type」を伝送しているIPデータフローと同一のIPデータフローであることを示す。なお、0xは、以降の文字列が16進数であることを示す。

10

【0059】

「location_type」が「0x01」であるときは、ロケーションが、UDP/IP（バージョン4）であることを示す。「location_type」が「0x02」であるときは、ロケーションが、UDP/IP（バージョン6）であることを示す。「location_type」が「0x03」であるときは、ロケーションが、ブロードキャストネットワークで配信されているMPEG-2TSであることを示す。「location_type」が「0x04」であるときは、ロケーションが、IPブロードキャストネットワークで配信されているMPEG-2TS内のエレメンタリーストリームであることを示す。

【0060】

20

「location_type」が「0x05」であるときは、ロケーションが、URL（Uniform Resource Locator）であることを示す。「location_type」が「0x06」であるときは、ロケーションが、URLで指定されたファイル中の連続した領域であることを示す。「location_type」が「0x07」であるときは、ロケーションが、この「location_type」を伝送している制御メッセージ内であることを示す。「location_type」が「0x08」であるときは、ロケーションが、この「location_type」を伝送しているデータパスにより伝送されている制御メッセージであることを示す。

【0061】

「location_type」が「0x09」であるときは、この「location_type」を伝送しているUDP/IPのデータフローにより伝送されている制御メッセージであることを示す。「location_type」が「0x0A」であるときは、UDP/IP（バージョン4）のデータフローにより伝送されている制御メッセージであることを示す。「location_type」が「0x0B」であるときは、UDP/IP（バージョン6）のデータフローにより伝送されている制御メッセージであることを示す。「location_type」が「0x0C」であるときは、IP（バージョン6）のブロードキャストネットワークで配信されているMPEG-2TS内のエレメンタリーストリームであることを示す。

30

【0062】

次のNo. 3の「if (location_type == 0x00) {」は、No. 2の「location_type」の値が「0x00」であるときに、No. 4の領域が配置されることを示す。次に配置されているNo. 4の「packet_id」は、MMTパケットのパケットIDを示す16ビットの領域である。

40

【0063】

次のNo. 5の「} else if (location_type == 0x01) {」は、No. 2の「location_type」の値が「0x01」であるときに、No. 9までの領域が配置されることを示す。次に配置されているNo. 6の「ipv4_src_addr」は、送信元のIPv4のアドレスを示す32ビットの領域である。次に配置されているNo. 7の「ipv4_dst_addr」は、宛先のIPv4のアドレスを示す32ビットの領域である。次に配置されているNo. 8の「dst_port」

50

は、宛先のポート番号を示す16ビットの領域である。次に配置されているNo. 9の「packet_id」は、MMTパケットのパケットIDを示す16ビットの領域である。

【0064】

次のNo. 10の「} else if (location_type == 0x02) {」は、No. 2の「location_type」の値が「0x02」であるときに、No. 14までの領域が配置されることを示す。次に配置されているNo. 11の「ipv6_src_addr」は、送信元のIPv6のアドレスを示す128ビットの領域である。次に配置されているNo. 12の「ipv6_dst_addr」は、宛先のIPv6のアドレスを示す128ビットの領域である。次に配置されているNo. 13の「dst_port」は、宛先のポート番号を示す16ビットの領域である。次に配置されているNo. 14の「packet_id」は、MMTパケットのパケットIDを示す16ビットの領域である。

【0065】

次のNo. 15の「} else if (location_type == 0x03) {」は、No. 2の「location_type」の値が「0x03」であるときに、No. 19までの領域が配置されることを示す。次に配置されているNo. 16の「network_id」は、MPEG2-TSを伝送しているブロードキャストネットワークのIDを示す16ビットの領域である。次に配置されているNo. 17の「MPEG2_transport_stream_id」は、MPEG2-TSのIDである。次に配置されているNo. 18の「reserved」は、3ビットの予約領域である。次に配置されているNo. 19の「MPEG2_PID」は、エレメンタリーストリームを伝送しているMPEG2-TSパケットのPIDである。

【0066】

次のNo. 20の「} else if (location_type == 0x04) {」は、No. 2の「location_type」の値が「0x04」であるときに、図10におけるNo. 25までの領域が配置されることを示す。次に配置されているNo. 21の「ipv6_src_addr」は、送信元のIPv6のアドレスを示す128ビットの領域である。次に配置されているNo. 22の「ipv6_dst_addr」は、宛先のIPv6のアドレスを示す128ビットの領域である。次に配置されているNo. 23の「dst_port」は、宛先のポート番号を示す16ビットの領域である。次に配置されている図10におけるNo. 24の「reserved」は、3ビットの予約領域である。次に配置されているNo. 25の「MPEG2_PID」は、エレメンタリーストリームを伝送しているMPEG2-TSパケットのPIDである。

【0067】

次のNo. 26の「} else if (location_type == 0x05) {」は、No. 2の「location_type」の値が「0x05」であるときに、No. 30までの領域が配置されることを示す。次に配置されているNo. 27の「URL_length」は、URLのバイト数N1を示す8ビットの領域である。次のNo. 28の「for (i = 0; i < N1; i++) {」は、以降、No. 30までの領域をN1回、すなわちURLのバイト数分繰り返すことを示す。次に配置されているNo. 29の「URL_byte」は、URL中の1バイトを示す領域である。次のNo. 30の「}」は、No. 28の「{」に対応する。

【0068】

次のNo. 31の「} else if (location_type == 0x06) {」は、No. 2の「location_type」の値が「0x06」であるときに、No. 37までの領域が配置されることを示す。次に配置されているNo. 32の「URL_length」は、URLのバイト数N2を示す8ビットの領域である。次のNo. 33の「for (i = 0; i < N2; i++) {」は、以降、No. 35までの領域をN2回、すなわちURLのバイト数分繰り返すことを示す。次に配置されているNo. 34の

10

20

30

40

50

「URL_byte」は、URL中の1バイトを示す領域である。次のNo. 35の「}」は、No. 33の「{」に対応する。次に配置されているNo. 36の「byte_offset」は、URLで示されるファイルの先頭からのオフセット位置を示す16ビットの領域である。次に配置されているNo. 37の「length」は、オフセット位置からのバイト数を示す16ビットの領域である。

【0069】

次のNo. 38の「} else if (location_type == 0x07) {」は、No. 2の「location_type」の値が「0x07」であるときに、何も配置されないことを示す。

【0070】

次のNo. 39の「} else if (location_type == 0x08) {」は、No. 2の「location_type」の値が「0x08」であるときに、No. 40の領域が配置されることを示す。次に配置されているNo. 40の「message_id」は、制御メッセージのIDを示す8ビットの領域である。

【0071】

次のNo. 41の「} else if (location_type == 0x09) {」は、No. 2の「location_type」の値が「0x09」であるときに、No. 43までの領域が配置されることを示す。次に配置されているNo. 42の「packet_id」は、MMTパケットのパケットIDを示す16ビットの領域である。次に配置されているNo. 43の「message_id」は、制御メッセージのIDを示す8ビットの領域である。

【0072】

次のNo. 44の「} else if (location_type == 0x0A) {」は、No. 2の「location_type」の値が「0x0A」であるときに、図11におけるNo. 49までの領域が配置されることを示す。次に配置されているNo. 45の「ipv4_src_addr」は、送信元のIPv4のアドレスを示す32ビットの領域である。次に配置されているNo. 46の「ipv4_dst_addr」は、宛先のIPv4のアドレスを示す32ビットの領域である。次に配置されている図11におけるNo. 47の「dst_port」は、宛先のポート番号を示す16ビットの領域である。次に配置されているNo. 48の「packet_id」は、MMTパケットのパケットIDを示す16ビットの領域である。次に配置されているNo. 49の「message_id」は、制御メッセージのIDを示す8ビットの領域である。

【0073】

次のNo. 50の「} else if (location_type == 0x0B) {」は、No. 2の「location_type」の値が「0x0B」であるときに、No. 55までの領域が配置されることを示す。次に配置されているNo. 51の「ipv6_src_addr」は、送信元のIPv6のアドレスを示す128ビットの領域である。次に配置されているNo. 52の「ipv6_dst_addr」は、宛先のIPv6のアドレスを示す128ビットの領域である。次に配置されているNo. 53の「dst_port」は、宛先のポート番号を示す16ビットの領域である。次に配置されているNo. 54の「packet_id」は、MMTパケットのパケットIDを示す16ビットの領域である。次に配置されているNo. 55の「message_id」は、制御メッセージのIDを示す8ビットの領域である。

【0074】

次のNo. 56の「} else if (location_type == 0x0C) {」は、No. 2の「location_type」の値が「0x0C」であるときに、No. 62までの領域が配置されることを示す。次に配置されているNo. 57の「ipv4_src_addr」は、送信元のIPv4のアドレスを示す32ビットの領域である。次に配置されているNo. 58の「ipv4_dst_addr」は、宛先のIPv4のアドレスを示す32ビットの領域である。次に配置されているNo. 58の「dst_

port」は、宛先のポート番号を示す16ビットの領域である。次に配置されているNo. 60の「reserved」は、3ビットの予約領域である。次に配置されているNo. 61の「MPEG_2_PID」は、エレメンタリーストリームを伝送しているMPEG2-TSパケットのPIDである。次のNo. 62の「}」は、No. 56の「{」に対応する。次のNo. 63の「}」は、No. 1の「{」に対応する。

【0075】

図12、図13は、パッケージリストテーブルのデータ構造を示す表である。該表において、No. は、該表における行の通し番号を示す欄である。データ構造は、パッケージリストテーブルのデータ構造を示す欄である。ビット数は、該データ構造のビット数を示す欄である。値は、該ビット数のデータ構造がとる値を示す欄である。ビット列表記は、該ビット数のデータ構造のデータ形式を示す欄である。ビット列表記のuimsbfは、unsigned integer most significant bit firstの略であり、符号無し整数、最上位ビットが先頭を意味する。また、ビット列表記のcharは、文字を意味する。

【0076】

図12において、No. 1の「Package_List_Table() {」は、以降、図13におけるNo. 30の「}」までが、パッケージリストテーブルのデータ構造であることを表す。当該テーブルの先頭に配置されているNo. 2の「table_id」は、パッケージリストテーブルのIDを示す8ビットの領域である。次に配置されているNo. 3の「version」は、パッケージリストテーブルのバージョン番号を示す8ビットの領域である。次に配置されているNo. 4の「length」は、当該領域以降のパッケージリストテーブルのバイト数を示す16ビットの領域である。

【0077】

次に配置されているNo. 5の「num_of_package」は、当該パッケージリストテーブルに参照情報を記載するパッケージの数N2を示す8ビットの領域である。次のNo. 6の「for(i=0; i<N2; i++) {」は、以降、No. 9までの領域をN2回、すなわち参照情報を記載するパッケージの数分繰り返すことを示す。次に配置されているNo. 7の「package_id」は、MPテーブルへの参照情報を記載するパッケージのパッケージIDを示す16ビットの領域である。なお、このNo. 7の「package_id」に変えて、図6のNo. 9からNo. 12と同様の領域を配置して、任意長のパッケージIDを記載できるようにしてもよい。次に配置されているNo. 8の「MMT_general_location_info()」は、MPテーブルへの参照情報であり、図9から図11にて示した参照情報を示す領域である。パッケージリストテーブルは、これら、No. 7とNo. 8とにより、パッケージIDとMPテーブルへの参照情報（パケットID）とを対応付けている。次のNo. 9の「}」は、No. 6の「{」に対応する。

【0078】

次に配置されているNo. 10の「num_of_ip_delivery」は、この参照情報を記載するIPサービスの数N3を示す8ビットの領域である。次のNo. 11の「for(i=0; i<N3; i++) {」は、以降、図13におけるNo. 29までの領域をN3回、すなわち参照情報を記載するIPサービスの数分繰り返すことを示す。次に配置されているNo. 12の「transport_file_id」は、IPサービスで伝送するファイルを一意に識別するためのラベルを示す32ビットの領域である。次に配置されているNo. 13の「location_type」は、参照情報のタイプを示す8ビットの領域である。「location_type」が「0x01」であるときは、IPv4のデータフローであることを示す。「location_type」が「0x02」であるときは、ロケーションが、UDP/IP（バージョン6）であることを示す。

【0079】

次のNo. 14の「if(location_type==0x01) {」は、No. 13の「location_type」の値が「0x01」であるときに、No. 17ま

10

20

30

40

50

での領域が配置されることを示す。次に配置されているNo. 15の「i p v 4 _ s r c _ a d d r」は、送信元のI P v 4のアドレスを示す32ビットの領域である。次に配置されているNo. 16の「i p v 4 _ d s t _ a d d r」は、宛先のI P v 4のアドレスを示す32ビットの領域である。次に配置されているNo. 17の「d s t _ p o r t」は、宛先のポート番号を示す16ビットの領域である。

【0080】

次のNo. 18の「i f (l o c a t i o n _ t y p e == 0 x 0 2) {」は、No. 13の「l o c a t i o n _ t y p e」の値が「0 x 0 2」であるときに、No. 21までの領域が配置されることを示す。次に配置されているNo. 19の「i p v 6 _ s r c _ a d d r」は、送信元のI P v 6のアドレスを示す128ビットの領域である。次に配置されているNo. 20の「i p v 6 _ d s t _ a d d r」は、宛先のI P v 6のアドレスを示す128ビットの領域である。次に配置されているNo. 21の「d s t _ p o r t」は、宛先のポート番号を示す16ビットの領域である。

【0081】

次の図13におけるNo. 22の「i f (l o c a t i o n _ t y p e == 0 x 0 5) {」は、図12におけるNo. 13の「l o c a t i o n _ t y p e」の値が「0 x 0 5」であるときに、No. 27までの領域が配置されることを示す。次に配置されているNo. 23の「U R L _ l e n g t h」は、URLのバイト数N1を示す8ビットの領域である。次のNo. 24の「f o r (i = 0 ; i < N 1 ; i + +) {」は、以降、No. 26までの領域をN1回、すなわちURLのバイト数分繰り返すことを示す。次に配置されているNo. 25の「U R L _ b y t e」は、URL中の1バイトを示す領域である。次のNo. 26の「}」は、No. 24の「{」に対応し、次のNo. 27の「}」は、No. 22の「{」に対応する。

【0082】

次に配置されているNo. 28の「d e s c r i p t o r s ()」は、IPサービスに関する記述子を示す領域である。次のNo. 29の「}」は、図12におけるNo. 11の「{」に対応し、次のNo. 30の「}」は、図12におけるNo. 1の「{」に対応する。

【0083】

図14は、指定取得部72、受信制御部73、制御メッセージ処理部74の動作を説明するフローチャートである。まず、指定取得部72は、所望のサービスの指定を取得する(Sa1)。次に、受信制御部73は、該所望のサービスを伝送しているIPデータフローの受信を、受信部70に指示する(Sa2)。なお、サービスと、IPデータフローとの対応は、受信制御部73が予め記憶していてもよいし、AMT (Address Map Table) など、該対応を示す情報がサービス提供側から配信されてもよい(非特許文献1、2参照)。

【0084】

受信部70が、指示に従いIPデータフローを受信すると、分離部71は、該IPデータフローからパケットIDが「0」のMMTパケットに格納されているPAメッセージを分離する。制御メッセージ処理部74は、分離部71が分離した該PAメッセージを取得する(Sa3)。次に、制御メッセージ処理部74は、取得したPAメッセージ内のMPテーブルを取得し(Sa4)、該MPTテーブルのパッケージIDが、ステップSa1にて指定取得部72が取得した所望のサービスのパッケージIDと一致しているか否かを判定する(Sa5)。

【0085】

なお、制御メッセージ処理部74は、例えば、サービスと、パッケージIDとの対応付けを予め記憶しておき、該対応付けを参照することで、所望のサービスのパッケージIDを取得してもよい。あるいは、サービスとパッケージIDとの対応付けを示す情報がサービス提供側から配信されており、制御メッセージ処理部74は、該情報を参照することで、所望のサービスのパッケージIDを取得してもよい。

【0086】

ステップS a 5にて一致していないと判定したときは（S a 5－No）、ステップS a 1 1に進む。制御メッセージ処理部7 4は、ステップS a 3にて取得したP Aメッセージ内のパッケージリストテーブルを取得する（S a 1 1）。次に、制御メッセージ処理部7 4は、取得したパッケージリストテーブルを参照して、所望のサービスのパッケージI Dと対応付けられたパケットI Dを取得する（S a 1 2）。

【0087】

制御メッセージ処理部7 4は、取得したパケットI DのP Aメッセージの分離を分離部7 4に指示する。分離部7 4は、指示されたパケットI DのP Aメッセージを、受信部7 0が受信しているI Pデータフローから分離する。制御メッセージ処理部7 4は、分離されたP Aメッセージを取得する。すなわち、制御メッセージ処理部7 4は、ステップS a 1 2にて取得したパケットI DのP Aメッセージを取得する（S a 1 3）。

10

【0088】

次に、制御メッセージ処理部7 4は、取得したP Aメッセージ内のM Pテーブルを取得し（S a 1 4）、ステップS a 6に進む。一方、ステップS a 5にて一致していると判定したときは（S a 5－Yes）、そのままステップS a 6に進む。

【0089】

ステップS a 6では、制御メッセージ処理部7 4は、取得したM Pテーブルに記載されたアセットの分離を分離部7 1に指示する。分離部7 1は、該指示にしたがい、分離したアセットのうち、映像ストリームは、映像デコード部7 5に入力し、音声ストリームは、音声デコード部7 6に入力し、データストリームは、データレンダリング部7 7に入力する。これにより、出力装置1 8が、該パッケージの映像の表示や、音声の出力を行うようになる。

20

【0090】

所望のサービスのM Pテーブルは、受信しているI Pデータフローにより繰り返し伝送されてくるので、制御メッセージ処理部7 4は、該M Pテーブルを監視する（S a 7）。制御メッセージ処理部7 4は、伝送されてきたM Pテーブルに、以前のものからの変更があるか否かを判定する（S a 8）。変更があると判定したときは（S a 8－Yes）、制御メッセージ処理部7 4は、M Pテーブルを再取得し（S a 9）、ステップS a 6に戻る。

30

【0091】

ステップS a 8にて、変更がないと判定したときは（S a 8－No）、指定取得部7 2が、所望のサービスが変更されたか否かを判定する（S a 1 0）。所望のサービスが変更されていないときは（S a 1 0－No）、ステップS a 7に戻る。また、所望のサービスが変更されているときは、ステップS a 1に戻る。

【0092】

このように、配信システム1 0は、制御メッセージ生成装置1 1と、配信装置1 6と、受信装置1 7とを有する。制御メッセージ生成装置1 1は、第1のパッケージを構成するアセット各々への参照情報を含む第1のM Pテーブル（第1のテーブル）と、第2のM Pテーブル（第2のテーブル）への参照情報を含むパッケージリストテーブルとを含む第1のP Aメッセージ（第1の制御メッセージ）を生成する。また、制御メッセージ生成装置1 1は、第2のM Pテーブルを含む第2のP Aメッセージ（第2の制御メッセージ）を生成する。配信装置1 6は、制御メッセージ生成装置1 1が生成した第1および第2のP Aメッセージを、I Pデータフロー（データフロー）を用いて配信する。受信装置1 7は、該データフローを受信する。

40

【0093】

これにより、配信システム1 0は、受信装置1 7が、第1のパッケージについては、第1の制御メッセージのみを取得すれば、すなわち、1ホップで受信可能となるようにすることができる。また、第2のパッケージについては、第1の制御メッセージと、第2の制御メッセージとを取得すれば、すなわち、2ホップで受信可能となるようにすることができる。

50

きる。したがって、パッケージを受信可能になるまでのホップ数の期待値を抑えることが出来る。

【0094】

また、受信装置17は、指定取得部72と、受信部70と、分離部71と、制御メッセージ処理部74とを具備する。指定取得部72は、所望のパッケージを指定する情報を取得する。受信部70は、IPデータフローを受信する。分離部71は、IPデータフローから第1のPAメッセージを分離する。制御メッセージ処理部74は、第1のPAメッセージに含まれる第1のMPテーブルを参照して、第1のパッケージが所望のパッケージと一致しているときは、第1のMPパッケージの分離を分離部71に指示する。

これにより、受信装置17は、第1のパッケージを1ホップで受信することができる。

10

【0095】

また、制御メッセージ処理部74は、第1のPAメッセージに含まれる第1のMPテーブルを参照して、第1のパッケージが所望のパッケージと一致していないときは、第1のPAメッセージに含まれるパッケージリストテーブルを参照して、所望のパッケージに対応する第2のMPテーブルへの参照情報を取得し、取得した参照情報を用いて、所望のパッケージに対応する第2のMPテーブルを含む第2のPAメッセージの分離を分離部71に指示する。

これにより、受信装置17は、第2のパッケージを2ホップで受信することができる。

【0096】

なお、上述の実施形態において、IPデータフロー中に、同時に含まれるパッケージが一つのときは、パケットIDが「0」のPAメッセージに、パッケージリストテーブルを含めなくてもよいし、IPデータフロー中に含まれる一つのパッケージに関する情報のみを含むパッケージリストテーブルを含めてもよいし、「num_of_package」の値を「0」にし、パッケージに関する参照情報を含まないパッケージリストテーブルを生成するようにしてもよい。

20

【0097】

また、上述の実施形態において、MPテーブルと、パッケージリストテーブルとは、PAメッセージに格納されるとして説明したが、その他の制御メッセージに格納されてもよい。ただし、MPテーブルとともに、パッケージリストテーブルを格納する制御メッセージは、パケットIDが「0」のPAメッセージであることが望ましい。パケットIDが「0」のPAメッセージは、受信装置17が最初を取得する制御メッセージとするからである。

30

【0098】

[実施形態の変形例1]

上述の実施形態では、制御メッセージ生成装置11は、パッケージリストテーブルを、パケットIDが「0」のPAメッセージにのみ格納しているが、その他のPAメッセージにも格納するようにしてもよい。図15は、変形例1におけるIPデータフローの構成を示す概念図である。同図において、図4の各部に対応する部分には、同一の符号を付し、説明を省略する。

【0099】

IPデータフローDFaには、図4のIPデータフローDFとは、PAメッセージPA1に変えて、PAメッセージPA2を含む点が異なる。PAメッセージPA2は、MPテーブルMP2に加えて、パッケージリストテーブルPL2を含む。このパッケージリストテーブルPL2には、MPテーブルMP2以外のMPテーブルへの参照情報を少なくとも含んでいる。なお、パッケージリストテーブルPL1、PL2ともに、IPデータフローDFaにて伝送されている全てのMPテーブルへの参照情報を含むようにしてもよい。

40

【0100】

[実施形態の変形例2]

以下、図面を参照して、実施形態の変形例2について説明する。本変形例は、上述の実施形態と、MPテーブル、参照情報(MMT_general_location_in

50

f o ())、パッケージリストテーブルのデータ構造が異なる。図16、図17、図18は、本変形例におけるMPテーブルのデータ構造を示す表である。これらの図で示すデータ構造は、図6、図7、図8に示すデータ構造とは、以下の4点が異なる。

【0101】

1点目は、本変形例のMPテーブルにおいて、図16のNo. 2の「table_id」の値が「0x20」であることである。2点目は、本変形例のMPテーブルが、図7のNo. 23からNo. 28に代えて、図17のNo. 23からNo. 28を有する点である。3点目は、本変形例のMPテーブルが、図7のNo. 28に代えて、図17のNo. 29に「asset_type」を有する点である。4点目は、本変形例のMPテーブルが、図7のNo. 30からNo. 36を有さない点である。

10

【0102】

以下、上記の2点目について説明する。図17のNo. 23の「identifier_type」は、識別子タイプともいい、以降に格納されるMMTPパケットフローのID体系を示す8ビットの領域である。ここでは、「identifier_type」の値は、アセットIDが格納されることを示す「0x00」である。No. 24の「asset_id_scheme」は、アセットID形式ともいい、以降に格納されるアセットIDの形式を示す32ビットの領域である。No. 25の「asset_id_length」は、アセットID長ともいい、以降に格納されるアセットIDバイトの長さをバイト単位で示す8ビットの領域である。次のNo. 26の「for (i=0; i<N4; i++) {」は、以降、No. 28までの領域をN4回、すなわちアセットID長の数だけ繰り返すことを示す。次に配置されているNo. 27の「asset_id_byte」は、アセットIDバイトともいい、アセットID中の1バイトを示す領域である。次のNo. 28の「}」は、No. 26の「{」に対応する。

20

【0103】

以下、上記の3点目について説明する。図17のNo. 29の「asset_type」は、アセットタイプともいい、No. 27のアセットIDバイトで示されたアセットの種類を示す32ビットの領域である。「asset_type」は、値が「hvc1」であるときは、アセットの種類がITU-T勧告H. 265で規定されるHEVCであることを示す。また「asset_type」は、値が「mp4a」であるときは、アセットの種類がISO/IEC 14496-3の音声であることを示す。

30

【0104】

また「asset_type」は、値が「stpp」であるときは、アセットの種類がタイムドテキスト（字幕・文字スーパー）であることを示す。また「asset_type」は、値が「aapp」であるときは、アセットの種類がアプリケーションであることを示す。また「asset_type」は、値が「asgd」であるときは、アセットの種類が同期型汎用データであることを示す。また「asset_type」は、値が「aagd」であるときは、アセットの種類が非同期型汎用データであることを示す。

【0105】

図19、図20は、本変形例における参照情報(MMT_general_location_info())のデータ構造を示す表である。これらの図で示すデータ構造は、図9、図10、図11に示すデータ構造とは、以下の4点が異なる。図10のNo. 31からNo. 46および図11のNo. 47からNo. 62を有しない点が異なる。

40

【0106】

図21、図22は、本変形例におけるパッケージリストテーブルのデータ構造を示す表である。これらの図で示すデータ構造は、図12、図13に示すデータ構造とは、以下の2点が異なる。1点目は、本変形例におけるパッケージリストテーブルが、図12のNo. 7に代えて、図21のNo. 7からNo. 10を有する点である。図21のNo. 7の「MMT_package_id_length」は、パッケージID長ともいい、以降に格納されるパッケージIDバイトの長さをバイト単位で示す8ビットの領域である。次のNo. 8の「for (j=0; j<M; j++) {」は、No. 9の領域をM回、すな

50

わちパッケージIDバイトの長さだけ繰り返すことを示す。次に配置されているNo. 9の「MMT_package_id_byte」は、パッケージIDバイトともいい、パッケージID中の1バイトを示す8ビットの領域である。次のNo. 10の「}」は、No. 8の「{」に対応する。

【0107】

2点目は、本変形例におけるパッケージリストテーブルが、図13のNo. 28に代えて、図22のNo. 32からNo. 34を有する点である。図22のNo. 32の「descriptor_loop_length」は、記述子長ともいい、以降に格納される記述子の全バイト長を示す16ビットの領域である。次のNo. 33の「for(j=0; j<M; j++) {」は、No. 33の領域をM回、すなわち記述子長だけ繰り返すことを示す。次のNo. 34の「descriptor()」は、記述子領域ともいい、IPサービスの詳細な情報を示す記述子のための領域中の1バイトである。次のNo. 34の「}」は、No. 32の「{」に対応する。

【0108】

なお、上述の変形例1におけるMPテーブル、MMT_general_location_info()、パッケージリストテーブルのうち、いずれかまたは全てが、変形例2と同様にであってもよい。

また、上述の変形例2では、MPテーブル、MMT_general_location_info()、パッケージリストテーブルの全てが、上述の実施形態と異なっていたが、異なっているのは、MPテーブル、MMT_general_location_info()、パッケージリストテーブルの一部であってもよい。

【0109】

また、図1における制御メッセージ生成装置11、受信装置17の機能を実現するためのプログラムをコンピュータ読み取り可能な記録媒体に記録して、この記録媒体に記録されたプログラムをコンピュータシステムに読み込ませ、実行することによりこれらの装置を実現してもよい。なお、ここでいう「コンピュータシステム」とは、OSや周辺機器等のハードウェアを含むものとする。

【0110】

また、「コンピュータ読み取り可能な記録媒体」とは、フレキシブルディスク、光磁気ディスク、ROM、CD-ROM等の可搬媒体、コンピュータシステムに内蔵されるハードディスク等の記憶装置のことをいう。さらに「コンピュータ読み取り可能な記録媒体」とは、インターネット等のネットワークや電話回線等の通信回線を介してプログラムを送信する場合の通信線のように、短時間の間、動的にプログラムを保持するもの、その場合のサーバやクライアントとなるコンピュータシステム内部の揮発性メモリのように、一定時間プログラムを保持しているものも含むものとする。また上記プログラムは、前述した機能の一部を実現するためのものであっても良く、さらに前述した機能をコンピュータシステムにすでに記録されているプログラムとの組み合わせで実現できるものであっても良い。

【0111】

また、上述した図2における制御メッセージ生成装置11、図3における受信装置17の各機能ブロックは個別にチップ化してもよいし、一部、または全部を集積してチップ化してもよい。また、集積回路化の手法はLSIに限らず、専用回路、または汎用プロセッサで実現しても良い。ハイブリッド、モノリシックのいずれでも良い。一部は、ハードウェアにより、一部はソフトウェアにより機能を実現させても良い。

また、半導体技術の進歩により、LSIに代替する集積回路化等の技術が出現した場合、当該技術による集積回路を用いることも可能である。

【0112】

以上、この発明の実施形態を図面を参照して詳述してきたが、具体的な構成はこの実施形態に限られるものではなく、この発明の要旨を逸脱しない範囲の設計変更等も含まれる。

10

20

30

40

50

【符号の説明】

【0113】

- 10 … 配信システム
 11 … 制御メッセージ生成装置
 12 … 映像送出装置
 13 … 音声送出装置
 14 … データ送出装置
 15 … 多重装置
 16 … 配信装置
 17 … 受信装置
 18 … 出力装置
 21 … MPテーブル生成部
 22 … パッケージリストテーブル生成部
 23 … PAメッセージ生成部
 70 … 受信部
 71 … 分離部
 72 … 指定取得部
 73 … 受信制御部
 74 … 制御メッセージ処理部
 75 … 映像デコード部
 76 … 音声デコード部
 77 … データレンダリング部
 78 … 映像合成部
 79 … 音声合成部

10

20

【図1】

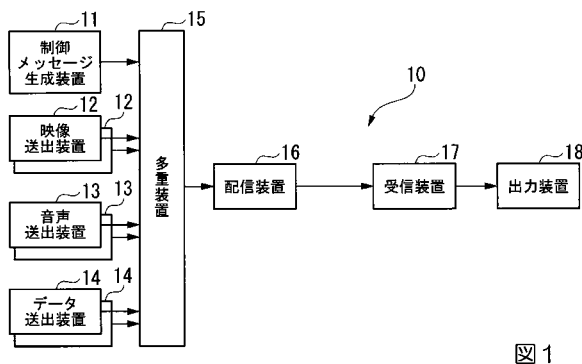


図1

【図2】

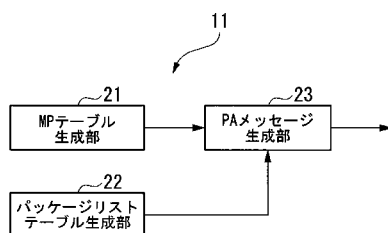


図2

【図3】

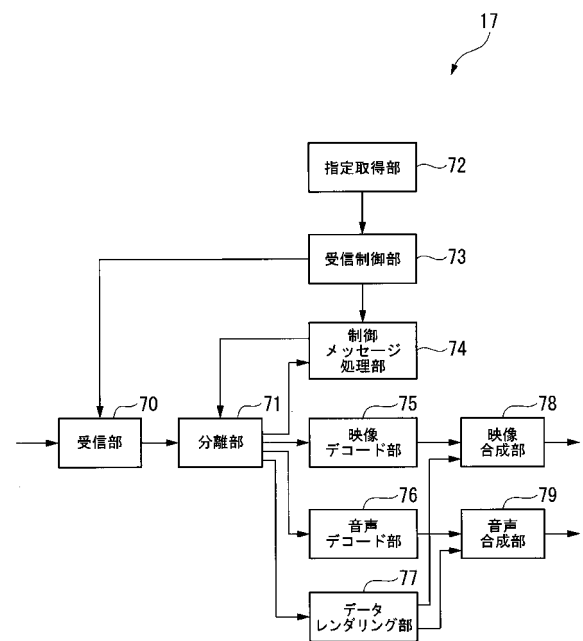


図3

【図 4】

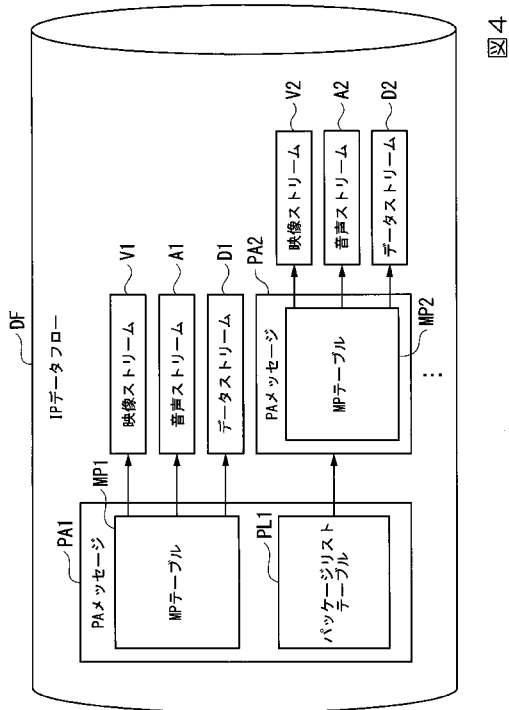


図4

【図 5】

No.	データ構造	ビット数	ビット列表記
1	PA_message() {		
2	message_id	16	uimsbf
3	version	8	uimsbf
4	length	32	uimsbf
5	extension{		
6	number_of_tables	8	uimsbf
7	for(i=0; i<N; i++) {		
8	table_id	8	uimsbf
9	table_version	8	uimsbf
10	table_length	16	uimsbf
11	}		
12	}		
13	message_payload{		
14	for(i=0; i<N; i++) {		
15	table()		
16	}		
17	}		
18	}		

図5

【図 6】

No.	データ構造	値	ビット数	ビット列表記
1	MPT() {			
2	table_id	0x40	8	uimsbf
3	version		8	uimsbf
4	length		16	uimsbf
5	reserved	'111111'	6	bslbf
6	MPT_mode		2	bslbf
7	if (table_id==SUBSET_O_MPT_TABLE_ID) {			
8	MNT_package_id{			
9	MNT_package_id_length	N1	8	uimsbf
10	for(i=0; i<N1; i++) {			
11	MNT_package_id_byte		8	uimsbf
12	}			
13	}			
14	MPT_descriptors{			
15	MPT_descriptors_length	N2	16	uimsbf
16	for(i=0; i<N2; i++) {			
17	MPT_descriptors_byte		8	uimsbf
18	}			
19	}			
20	}			

図6

【図 7】

No.	データ構造	値	ビット数	ビット列表記
21	number_of_assets	N3	8	uimsbf
22	for(i=0; i<N3; i++) {			
23	asset_id{			
24	asset_id_length	N4	8	uimsbf
25	for(j=0; j<N4; j++) {			
26	asset_id_byte		8	uimsbf
27	}			
28	mime_type		32	uimsbf
29	}			
30	if (asset_scheme_code== "GURL") {			
31	packet_id		32	uimsbf
32	number_of_codepoints	N8	8	uimsbf
33	for(k=0; k<N8; k++) {			
34	CodePoint()			
35	}			
36	}			
37	reserved	'111111'	7	bslbf
38	asset_clock_relation_flag		1	bslbf
39	if (asset_clock_relation_flag==1) {			
40	asset_clock_relation_id		8	uimsbf

図7

【図 8】

No.	データ構造	値	ビット 数	ビット 列表記
41	reserved	'111111'	7	bslbf
42	asset_timescale_flag		1	bslbf
43	if(asset_timescale_flag==1){			
44	asset_timescale		32	uimsbf
45	}			
46	}			
47	asset_location{			
48	location_count	N6	8	uimsbf
49	for(i=0;i<N6;i++){			
50	MMT_general_location_info()			
51	}			
52	}			
53	asset_descriptors{			
54	asset_descriptors_length	N5	16	uimsbf
55	for(j=0;j<N5;j++){			
56	asset_descriptors_byte		8	uimsbf
57	}			
58	}			
59	}			
60	}			

図 8

【図 9】

No.	データ構造	値	ビット 数	ビット 列表記
1	MMT_general_location_info(){			
2	location_type		8	uimsbf
3	if(location_type==0x00){			
4	packet_id		16	uimsbf
5	}else if(location_type==0x01){			
6	ipv4_src_addr		32	uimsbf
7	ipv4_dst_addr		32	uimsbf
8	dst_port		16	uimsbf
9	packet_id		16	uimsbf
10	}else if(location_type==0x02){			
11	ipv6_src_addr		128	uimsbf
12	ipv6_dst_addr		128	uimsbf
13	dst_port		16	uimsbf
14	packet_id		16	uimsbf
15	}else if(location_type==0x03){			
16	network_id		16	uimsbf
17	MPEG_2_transport_stream_id		16	uimsbf
18	reserved	'111'	3	bslbf
19	MPEG_2_PID		13	uimsbf
20	}else if(location_type==0x04){			
21	ipv6_src_addr		128	uimsbf
22	ipv6_dst_addr		128	uimsbf
23	dst_port		16	uimsbf

図 9

【図 10】

No.	データ構造	値	ビット 数	ビット 列表記
24	reserved	'111'	3	bslbf
25	MPEG_2_PID		13	uimsbf
26	}else if(location_type==0x05){			
27	URL_length	N1	8	uimsbf
28	for(i=0;i<N1;i++){			
29	URL_byte		8	char
30	}			
31	}else if(location_type==0x06){			
32	URL_length	N2	8	uimsbf
33	for(i=0;i<N2;i++){			
34	URL_byte		8	char
35	}			
36	byte_offset		16	uimsbf
37	length		16	uimsbf
38	}else if(location_type==0x07){			
39	}else if(location_type==0x08){			
40	message_id		8	uimsbf
41	}else if(location_type==0x09){			
42	packet_id		16	uimsbf
43	message_id		8	uimsbf
44	}else if(location_type==0x0A){			
45	ipv4_src_addr		32	uimsbf
46	ipv4_dst_addr		32	uimsbf

図 10

【図 11】

No.	データ構造	値	ビット 数	ビット 列表記
47	dst_port		16	uimsbf
48	packet_id		16	uimsbf
49	message_id		8	uimsbf
50	}else if(location_type==0x0B){			
51	ipv6_src_addr		128	uimsbf
52	ipv6_dst_addr		128	uimsbf
53	dst_port		16	uimsbf
54	packet_id		16	uimsbf
55	message_id		8	uimsbf
56	}else if(location_type==0x0C){			
57	ipv4_src_addr		32	uimsbf
58	ipv4_dst_addr		32	uimsbf
59	dst_port		16	uimsbf
60	reserved	'111'	3	bslbf
61	MPEG_2_PID		13	uimsbf
62	}			
63	}			

図 11

【図 1 2】

No.	データ構造	値	ビット 数	ビット 列表記
1	Package_List_Table() {			
2	table_id		8	uimsbf
3	version		8	uimsbf
4	length		16	uimsbf
5	num_of_package	N2	8	uimsbf
6	for (i=0; i<N2; i++) {			
7	package_id		16	uimsbf
8	MMT_general_location_info()			
9	}			
10	num_of_ip_delivery	N3	8	
11	for (i=0; i<N3; i++) {			
12	transport_file_id		32	uimsbf
13	location_type		8	uimsbf
14	if (location_type==0x01) {			
15	ipv4_src_addr		32	uimsbf
16	ipv4_dst_addr		32	uimsbf
17	dst_port		16	uimsbf
18	if (location_type==0x02) {			
19	ipv6_src_addr		128	uimsbf
20	ipv6_dst_addr		128	uimsbf
21	dst_port		16	uimsbf

図 1 2

【図 1 3】

No.	データ構造	値	ビット 数	ビット 列表記
22	if (location_type==0x05) {			
23	URL_length	N1	8	uimsbf
24	for (i=0; i<N1; i++) {			
25	URL_byte		8	char
26	}			
27	}			
28	descriptor()			
29	}			
30	}			

図 1 3

【図 1 4】

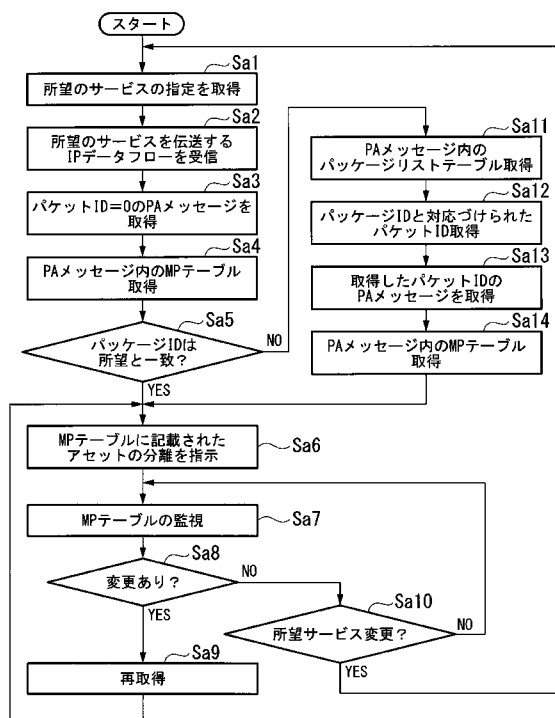


図 1 4

【図 1 5】

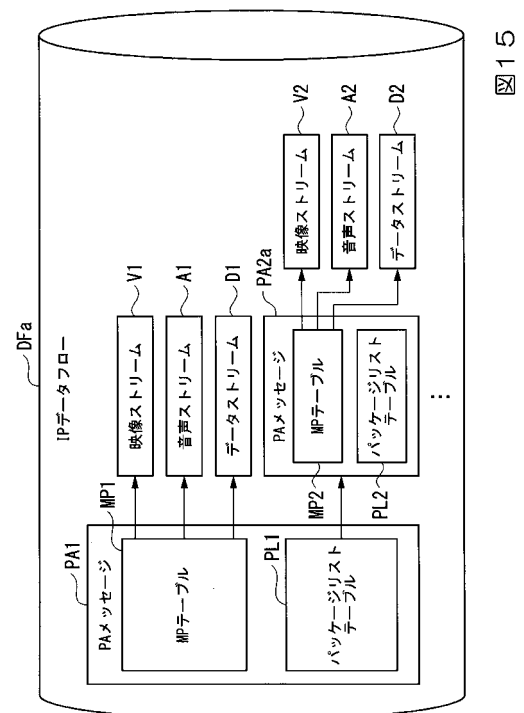


図 1 5

【図 16】

No.	データ構造	値	ビット数	ビット列表記
1	MPT() {			
2	table_id	0x20	8	uimsbf
3	version		8	uimsbf
4	length		16	uimsbf
5	reserved	'11 1111'	6	bslbf
6	MPT_mode		2	bslbf
7	if(table_id==SUBSET_0_MPT_TABLE_ID) {			
8	MMT_package_id {			
9	MMT_package_id_length	N1	8	uimsbf
10	for (i=0; i<N1; i++) {			
11	MMT_package_id_byte		8	uimsbf
12	}			
13	}			
14	MPT_descriptors {			
15	MPT_descriptors_length	N2	16	uimsbf
16	for (i=0; i<N2; i++) {			
17	MPT_descriptors_byte		8	uimsbf
18	}			
19	}			
20	}			

図 16

【図 17】

No.	データ構造	値	ビット数	ビット列表記
21	number_of_assets	N3	8	uimsbf
22	for (i=0; i<N3; i++) {			
23	identifier_type	0x00	8	uimsbf
24	asset_id_scheme		32	uimsbf
25	asset_id_length	N4	8	uimsbf
26	for (j=0; j<N4; j++) {			
27	asset_id_byte		8	uimsbf
28	}			
29	asset_type		32	char
30	reserved	'1111111'	7	bslbf
31	asset_clock_relation_flag		1	bslbf
32	if(asset_clock_relation_flag==1) {			
33	asset_clock_relation_id		8	uimsbf
34	reserved	'1111111'	7	bslbf
35	asset_timescale_flag		1	bslbf
36	if(asset_timescale_flag==1) {			
37	asset_timescale		32	uimsbf
38	}			
39	}			

図 17

【図 18】

No.	データ構造	値	ビット数	ビット列表記
40	asset_location {			
41	location_count	N6	8	uimsbf
42	for (i=0; i<N6; i++) {			
43	MMT_general_location_info ()			
44	}			
45	}			
46	asset_descriptors {			
47	asset_descriptors_length	N5	16	uimsbf
48	for (j=0; j<N5; j++) {			
49	asset_descriptors_byte		8	bslbf
50	}			
51	}			
52	}			
53	}			

図 18

【図 19】

No.	データ構造	値	ビット数	ビット列表記
1	MMT_general_location_info () {			
2	location_type		8	uimsbf
3	if(location_type==0x00) {			
4	packet_id		16	uimsbf
5	}else if(location_type==0x01) {			
6	ipv4_src_addr		32	uimsbf
7	ipv4_dst_addr		32	uimsbf
8	dst_port		16	uimsbf
9	packet_id		16	uimsbf
10	}else if(location_type==0x02) {			
11	ipv6_src_addr		128	uimsbf
12	ipv6_dst_addr		128	uimsbf
13	dst_port		16	uimsbf
14	packet_id		16	uimsbf
15	}else if(location_type==0x03) {			
16	network_id		16	uimsbf
17	MPEG_2_transport_stream_id		16	uimsbf
18	reserved	'111'	3	bslbf
19	MPEG_2_PID		13	uimsbf

図 19

【図 20】

No.	データ構造	値	ビット 数	ビット 列表記
20	}else if(location_type==0x04) {			
21	ipv6_src_addr		128	uimsbf
22	ipv6_dst_addr		128	uimsbf
23	dst_port		16	uimsbf
24	reserved	'111'	3	bslbf
25	MPEG_2_PID		13	uimsbf
26	}else if(location_type==0x05) {			
27	URL_length	N1	8	uimsbf
28	for (i=0; i<N1; i++) {			
29	URL_byte		8	char
30	}			
31	}			

図20

【図 21】

No.	データ構造	値	ビット 数	ビット 列表記
1	Package_List_Table() {			
2	table_id		8	uimsbf
3	version		8	uimsbf
4	length		16	uimsbf
5	num_of_package	N2	8	uimsbf
6	for (i=0; i<N2; i++) {			
7	MMT_package_id_length	M	8	uimsbf
8	for (j=0; j<M; j++) {			
9	MMT_package_id_byte		8	bslbf
10	}			
11	MMT_general_location_info()			
12	}			
13	num_of_ip_delivery	N3	8	
14	for (i=0; i<N3; i++) {			
15	transport_file_id		32	uimsbf
16	location_type		8	uimsbf
17	if (location_type==0x01) {			
18	ipv4_src_addr		32	uimsbf
19	ipv4_dst_addr		32	uimsbf
20	dst_port		16	uimsbf

図21

【図 22】

No.	データ構造	値	ビット 数	ビット 列表記
21	if(location_type==0x02) {			
22	ipv6_src_addr		128	uimsbf
23	ipv6_dst_addr		128	uimsbf
24	dst_port		16	uimsbf
25	if(location_type==0x05) {			
26	URL_length	N1	8	uimsbf
27	for (i=0; i<N1; i++) {			
28	URL_byte		8	char
29	}			
30	}			
31	descriptor_loop_length		16	uimsbf
32	for (j=0; j<M; j++) {			
33	descriptor()			
34	}			
35	}			
36	}			

図22

【手続補正書】

【提出日】平成30年10月10日(2018.10.10)

【手続補正 1】

【補正対象書類名】特許請求の範囲

【補正対象項目名】全文

【補正方法】変更

【補正の内容】

【特許請求の範囲】

【請求項 1】

第 1 のパッケージを構成するアセット各々への参照情報を含む第 1 のテーブルと、第 2 のパッケージを構成するアセット各々への参照情報を含む第 2 のテーブルとを生成するテーブル生成部と、

前記第 2 のテーブルへの参照情報を含むパッケージリストテーブルを生成するパッケージリストテーブル生成部と、

前記第 1 のテーブルおよび前記パッケージリストテーブルを含む第 1 の制御メッセージと、前記第 2 のテーブルを含む第 2 の制御メッセージとを生成する制御メッセージ生成部と、

を具備し、

前記第 2 のテーブルへの参照情報は、

前記第 2 のパッケージを示すパッケージ ID と、

前記第 2 のテーブルを含む前記第 2 の制御メッセージを格納するパケットのパケット ID と、

であり、

前記パッケージ ID の下位 16 ビットはサービス ID と同じ値である、

ことを特徴とする制御メッセージ生成装置。

フロントページの続き

特許法第30条第2項適用申請有り 第30条第2項適用、第12回情報科学技術フォーラム（FIT2013）講演論文集（平成25年8月20日）一般社団法人情報処理学会発行第I-050ページに発表

(72)発明者 大概 一博

東京都世田谷区砧一丁目10番11号 日本放送協会放送技術研究所内

Fターム(参考) 5C164 FA04 GA03 SB07S SB15P