

(19) 日本国特許庁(JP)

(12) 特許公報(B1)

(11) 特許番号

特許第5725235号
(P5725235)

(45) 発行日 平成27年5月27日 (2015. 5. 27)

(24) 登録日 平成27年4月10日 (2015. 4. 10)

(51) Int. Cl.	F I
HO 4 N 21/235 (2011. 01)	HO 4 N 21/235
HO 4 N 21/435 (2011. 01)	HO 4 N 21/435
HO 4 N 21/6543 (2011. 01)	HO 4 N 21/6543

請求項の数 8 (全 52 頁)

(21) 出願番号	特願2014-88630 (P2014-88630)	(73) 特許権者	000002185
(22) 出願日	平成26年4月22日 (2014. 4. 22)		ソニー株式会社
審査請求日	平成26年11月27日 (2014. 11. 27)		東京都港区港南1丁目7番1号
早期審査対象出願		(74) 代理人	100093241
			弁理士 官田 正昭
		(74) 代理人	100101801
			弁理士 山田 英治
		(74) 代理人	100095496
			弁理士 佐々木 榮二
		(74) 代理人	100086531
			弁理士 澤田 俊夫
		(74) 代理人	110000763
			特許業務法人大同特許事務所

最終頁に続く

(54) 【発明の名称】 受信装置及び受信方法、並びに、送信装置及び送信方法

(57) 【特許請求の範囲】

【請求項 1】

タイムド・メディア及びノンタイムド・メディアからなる各アセットに関するシグナリング・メッセージを生成するメッセージ生成部と、

前記シグナリング・メッセージを格納するパケットを所定のトランスポート方式で送信する送信部と、

を具備し、

前記ノンタイムド・メディアは、データ放送アプリケーションのアセットを構成するファイル又はディレクトリーに相当する複数のアイテムからなり、

前記メッセージ生成部は、ノンタイムド・メディア内のデータ放送アプリケーションの提示単位を識別する提示単位識別情報と、前記提示単位に含まれるアイテムを識別するアイテムタグ情報を示す第1のテーブルを含むシグナリング・メッセージを生成する、送信装置。

【請求項 2】

前記所定のトランスポート方式はMMTである、請求項1に記載の送信装置。

【請求項 3】

前記シグナリング・メッセージは、シグナリング・メッセージ上でアイテムを識別するアイテムタグ情報に対応するトランスポート・ストリーム上でアイテム識別するアイテム識別情報を示す第2のテーブルをさらに含む、

請求項 2 に記載の送信装置。

【請求項 4】

前記第 1 のテーブルは、前記アプリケーション提示単位にリンクする他のアプリケーション提示単位を識別するリンク先提示単位識別情報をさらに示す、

請求項 3 に記載の送信装置。

【請求項 5】

前記第 2 のテーブルは、アイテムを含むアセットを識別するアセット識別情報をさらに示す、

請求項 4 に記載の送信装置。

【請求項 6】

送信装置が備えるメッセージ生成部が、タイムド・メディア及びノンタイムド・メディアからなる各アセットに関するシグナリング・メッセージを生成するメッセージ生成ステップと、

前記送信装置が備える送信部が、前記シグナリング・メッセージを格納するパケットを所定のトランスポート方式で送信する送信ステップと、
を有し、

前記ノンタイムド・メディアは、データ放送アプリケーションのアセットを構成するファイル又はディレクトリーに相当する複数のアイテムからなり、

前記メッセージ生成ステップでは、ノンタイムド・メディア内のデータ放送アプリケーションの提示単位を識別する提示単位識別情報と、前記提示単位に含まれるアイテムを識別するアイテムタグ情報を示す第 1 のテーブルを含むシグナリング・メッセージを生成する、

送信方法。

【請求項 7】

タイムド・メディア及びノンタイムド・メディアからなる各アセットに関するシグナリング・メッセージが格納されている所定のトランスポート方式の伝送信号を受信する受信部と、

受信したシグナリング・メッセージに基づいてノンタイムド・メディアのキャッシュを制御する制御部と、

を具備し、

前記ノンタイムド・メディアは、データ放送アプリケーションのアセットを構成するファイル又はディレクトリーに相当する複数のアイテムからなり、

前記制御部は、受信したシグナリング・メッセージに含まれる、ノンタイムド・メディア内のデータ放送アプリケーションの提示単位を識別する提示単位識別情報と、前記提示単位に含まれるアイテムを識別するアイテムタグ情報を示す第 1 のテーブルに基づいて、提示単位でのアイテムのキャッシュを制御する、

受信装置。

【請求項 8】

受信装置が備える受信部が、タイムド・メディア及びノンタイムド・メディアからなる各アセットとノンタイムド・メディアに関するシグナリング・メッセージが格納されている所定のトランスポート方式の伝送信号を受信する受信ステップと、

前記受信装置が備える制御部が、受信したシグナリング・メッセージに基づいてノンタイムド・メディアのキャッシュを制御する制御ステップと、

を有し、

前記ノンタイムド・メディアは、データ放送アプリケーションのアセットを構成するファイル又はディレクトリーに相当する複数のアイテムからなり、

前記制御ステップでは、受信したシグナリング・メッセージに含まれる、ノンタイムド・メディア内のデータ放送アプリケーションの提示単位を識別する提示単位識別情報と、前記提示単位に含まれるアイテムを識別するアイテムタグ情報を示す第 1 のテーブルに基づいて、提示単位でのアイテムのキャッシュを制御する、

10

20

30

40

50

受信方法。

【発明の詳細な説明】

【技術分野】

【0001】

本明細書で開示する技術は、ファイルを受信する受信装置及び受信方法、並びに、ファイルを送信する送信装置及び送信方法に係り、例えば、所定のトランスポート方式で伝送されるファイルを受信する受信装置及び受信方法、並びに、送信装置及び送信方法に関する。

【背景技術】

【0002】

現在の放送システムでは、メディアのトランスポート方式として、MPEG-2 TS (Moving Picture Experts Group-2 Transport Stream)方式やRTP (Real Time Protocol)方式が広く使用されている(例えば、特許文献1を参照のこと)。次世代のデジタル放送方式として、MPEGで新たなメディア・トランスポート方式として規格化されたMMT (MPEG Media Transport) (例えば、非特許文献1を参照のこと)が検討されている。MMTでは異なる伝送路の組み合わせで利用することが容易であり、放送や通信の複数の伝送路に共通に用いることができる。

【0003】

MMT方式によれば、MMTパケット上で、ストリーム・メディアである映像や音声などのタイムド・メディア (Timed media) と、ファイルのようなノンタイムド・メディア (Non timed media) の両方を伝送することが可能である。

【先行技術文献】

【特許文献】

【0004】

【特許文献1】特開2013-153291号公報

【非特許文献】

【0005】

【非特許文献1】ISO/IEC FDIS 23008-1:2013(E) Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 1: MPEG media transport (MMT)

【発明の概要】

【発明が解決しようとする課題】

【0006】

本明細書で開示する技術の目的は、所定のトランスポート方式で伝送されるファイルを好適に受信することができる、優れた受信装置及び受信方法、並びに、所定のトランスポート方式でファイルを好適に送信することができる、優れた送信装置及び送信方法を提供することにある。

【課題を解決するための手段】

【0007】

本願は、上記課題を参酌してなされたものであり、請求項1に記載の技術は、所定のトランスポート方式の伝送信号を受信する受信部と、受信信号を各メディア・データとシグナリング情報にデマルチプレクスするデマルチプレクサーと、デマルチプレクスされたシグナリング情報に基づいて、デマルチプレクスされた少なくとも1つのメディア・データのキャッシュを制御する制御部と、を具備する受信装置である。

【0008】

10

20

30

40

50

本願の請求項 2 に記載の技術によれば、請求項 1 に記載の受信装置において、前記所定の
 のトランスポート方式は M M T である。

【 0 0 0 9 】

本願の請求項 3 に記載の技術によれば、請求項 1 に記載の受信装置において、伝送信号
 はタイムド・メディア及びノンタイムド・メディアを含み、前記制御部は、ノンタイムド
 ・メディアのキャッシュを制御するように構成されている。

【 0 0 1 0 】

本願の請求項 4 に記載の技術によれば、請求項 1 に記載の受信装置において、前記ノン
 タイムド・メディアは、データ放送アプリケーションのアセットを構成する複数のファイ
 ル・データからなり、前記制御部は、キャッシュの空き容量に応じて、ファイル・デー
 タのキャッシュを制御するように構成されている。

10

【 0 0 1 1 】

本願の請求項 5 に記載の技術によれば、請求項 4 に記載の受信装置の前記制御部は、受
 信したシグナリング情報に基づいて、データ放送アプリケーションを実行する際の階層的
 なファイル・データへのアクセス範囲と、各アクセス範囲におけるファイル・データのロ
 ケーション情報及びサイズを取得して、キャッシュを制御するように構成されている。

【 0 0 1 2 】

本願の請求項 6 に記載の技術によれば、請求項 4 又は 5 のいずれかに記載の受信装置の
 前記制御部は、以下の 4 通りの階層的なアクセス範囲 (a) ~ (d) とそのサイズを把握
 して、キャッシュの空き容量に応じていずれのアクセス範囲でキャッシュするかを決定す
 るように構成されている。

20

(a) アプリケーション実行時に直接参照するファイル・データ

(b) 同時に表示するアプリケーション提示単位を構成するファイル・データのリソース
 ・グループ

(c) 同時に表示するアプリケーション提示単位を構成するリソース・グループと、これ
 にリンクする他のアプリケーション提示単位を含む大きなリソース・グループ

(d) アプリケーション全体のリソース・グループ

【 0 0 1 3 】

本願の請求項 7 に記載の技術によれば、請求項 6 に記載の受信装置の前記制御部は、シ
 グナリング情報として、アプリケーションにおいてロケーションを示す U R L を放送など
 の M M T 伝送路上のロケーションにマッピングする第 1 の情報と、アプリケーションにお
 ける表示単位とリンク関係を示す第 2 の情報を受信して、前記 4 通りの階層的なアクセ
 ス範囲 (a) ~ (d) とそのサイズを把握するように構成されている。

30

【 0 0 1 4 】

本願の請求項 8 に記載の技術によれば、請求項 7 に記載の受信装置の前記受信部は、前
 記第 1 の情報として、アプリケーションの伝送方法とロケーションを示す U R L 情報を管
 理するアプリケーション情報テーブルと、アプリケーションにおいてロケーションを示す
 U R L とデータ・トランスミッション・メッセージ上の i t e m 識別情報との対応関係を
 管理するデータ・ロケーション・マネジメント・テーブルと、 i t e m 情報と、そのアイ
 テムを含むアセットの a s s e t 識別情報、そのアセットの M M T 伝送路上の M M T P
 パケットの d o w n l o a d _ i d 及び i t e m 識別情報との対応関係を管理するデータ・
 アセット・マネジメント・テーブルと、 a s s e t 識別情報と M M T 伝送路上でそのアセ
 ットを伝送するパケットの p a c k e t 識別情報の対応関係を管理する M M T パッケージ
 ・テーブルを受信するように構成されている。

40

【 0 0 1 5 】

本願の請求項 9 に記載の技術によれば、請求項 7 又は 8 のいずれかに記載の前記受信部
 は、前記第 2 の情報として、コンテンツに含まれるアプリケーション提示単位の識別情報
 、アプリケーション提示単位に含まれるすべてのアイテムの i t e m 識別情報と、これに
 リンクする他のアプリケーション提示単位の識別情報を管理するデータ・コンテンツ・マ
 ネジメント・テーブルを受信するように構成されている。

50

【 0 0 1 6 】

本願の請求項 1 0 に記載の技術によれば、請求項 1 乃至 9 のいずれかに記載の受信装置の前記制御部は、`pre fetch`（先読み）に状態が制御されたアプリケーションに関するシグナリング情報を受信したことに応答して、キャッシュ処理を実行するように構成されている。

【 0 0 1 7 】

また、本願の請求項 1 1 に記載の技術は、
所定のトランスポート方式の伝送信号を受信する受信ステップと、
受信信号を各メディア・データとシグナリング情報にデマルチプレクスするデマルチプレクス・ステップと、
デマルチプレクスされたシグナリング情報に基づいて、デマルチプレクスされた少なくとも一つのメディア・データのキャッシュを制御する制御ステップと、
を有する受信方法である。

10

【 0 0 1 8 】

また、本願の請求項 1 2 に記載の技術は、
タイムド・メディア及びノンタイムド・メディアと、受信側でノンタイムド・メディアのキャッシュ処理に利用可能な情報を含むシグナリング・メッセージをマルチプレクスするマルチプレクサーと、
マルチプレクスされたパケットを所定のトランスポート方式で送信する送信部と、
を具備する送信装置である。

20

【 0 0 1 9 】

本願の請求項 1 3 に記載の技術によれば、請求項 1 2 に記載の送信装置において、前記所定のトランスポート方式は M M T である。

【 0 0 2 0 】

本願の請求項 1 4 に記載の技術によれば、請求項 1 2 又は 1 3 のいずれかに記載の送信装置において、前記ノンタイムド・メディアは、データ放送アプリケーションのアセットを構成する複数のファイル・データからなり、前記シグナリング・メッセージは、アプリケーションにおいてロケーションを示す U R L を放送などの M M T 伝送路上のロケーションにマッピングする第 1 の情報と、アプリケーションにおける表示単位とリンク関係を示す第 2 の情報を含む。

30

【 0 0 2 1 】

本願の請求項 1 5 に記載の技術によれば、請求項 1 4 に記載の送信装置において、前記第 1 の情報は、アプリケーションの伝送方法とロケーションを示す U R L 情報を管理するアプリケーション情報テーブルと、アプリケーションにおいてロケーションを示す U R L とデータ・トランスミッション・メッセージ上の `item` 識別情報との対応関係を管理するデータ・ロケーション・マネジメント・テーブルと、`item` 識別情報と、そのアイテムを含むアセットの `asset` 識別情報、そのアセットの M M T 伝送路上の M M T P パケットの `download_id` 及び `item` 識別情報との対応関係を管理するデータ・アセット・マネジメント・テーブルと、`asset` 識別情報と M M T 伝送路上でそのアセットを伝送するパケットの `packet` 識別除法の対応関係を管理する M M T パッケージ・テーブルを含む。

40

【 0 0 2 2 】

本願の請求項 1 6 に記載の技術によれば、請求項 1 4 又は 1 5 のいずれかに記載の送信装置において、前記第 2 の情報は、コンテンツに含まれるアプリケーション提示単位の識別情報、アプリケーション提示単位に含まれるすべてのアイテムの `item` 識別情報と、これにリンクする他のアプリケーション提示単位の識別情報を管理するデータ・コンテンツ・マネジメント・テーブルを含む。

【 0 0 2 3 】

また、本願の請求項 1 7 に記載の技術は、
タイムド・メディア及びノンタイムド・メディアと、受信側でノンタイムド・メディア

50

のキャッシュ処理に利用可能な情報を含むシグナリング・メッセージをマルチプレクスするマルチプレクス・ステップと、

マルチプレクスされたパケットを所定のトランスポート方式で送信する送信ステップと、
を有する送信方法である。

【発明の効果】

【0024】

本明細書で開示する技術によれば、所定のトランスポート方式で伝送されるファイルを好適に受信することができる、優れた受信装置及び受信方法、並びに、所定のトランスポート方式でファイルを好適に送信することができる、優れた送信装置及び送信方法を提供することができる。

10

【0025】

なお、本明細書に記載された効果は、あくまでも例示であり、本発明の効果はこれに限定されるものではない。また、本発明が、上記の効果以外に、さらに付加的な効果を奏する場合もある。

【0026】

本明細書で開示する技術のさらに他の目的、特徴や利点は、後述する実施形態や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【図面の簡単な説明】

【0027】

20

【図1】図1は、本明細書で開示する技術を適用したデジタル放送システム10の構成例を模式的に示した図である。

【図2】図2は、MMTを適用した放送信号のスタック・モデル200を示した図である。

【図3】図3は、図2に示した放送信号を送出する放送送出システム11の構成例を示した図である。

【図4】図4は、図2に示した放送信号を受信する受信機12の構成例を示した図である。

【図5】図5は、MMT方式に従って放送送出システム11からRF伝送路に送出される放送信号(パッケージ)500のイメージを示した図である。

30

【図6】図6は、MMTパケットのヘッダーの構成例を示した図である。

【図7】図7は、ノンタイムド・メディアを伝送するMMTPパケットの場合の拡張ヘッダー700の構成例を示した図である。

【図8】図8は、MPUモードの場合のMMTPペイロード800の構成例を示した図である。

【図9】図9は、ペイロードにタイムド・メディアを配置したMFUのDU_Header900の構成例を示した図である。

【図10】図10は、ペイロードにノンタイムド・メディアを配置したMFUのDU_Header1000の構成例を示した図である。

【図11】図11は、ノンタイムド・メディアのデータを伝送する際のパケット構成例を示した図である。

40

【図12】図12は、PAメッセージ1201と、PAメッセージに含まれるMPテーブル1202の構成例を示した図である。

【図13】図13は、PAメッセージ1300のシンタックス例を示した図である。

【図14】図14は、PAメッセージに含まれるパラメーターの説明を示した図である。

【図15】図15は、MPテーブル(MPT)のシンタックス例(前半部分)を示した図である。

【図16】図16は、MPテーブルのシンタックス例を示した図である。

【図17】図17は、MPテーブルに含まれる各パラメーターを説明した図である。

【図18】図18は、M2セクション・メッセージ1800の構成例を示した図である。

50

【図19】図19は、M2セクション・メッセージで伝送されるMH AI (Application Information) テーブル (MH AIT) 1900の構成例を示した図である。

【図20】図20は、アプリケーション情報記述子2000の構成例を示した図である。

【図21】図21は、アプリケーション情報記述子に含まれるパラメータの説明を示した図である。

【図22】図22は、伝送プロトコル記述子2200の構成例を示した図である。

【図23】図23は、HTTP/HTTPS、MMT、及びノンタイムド伝送に共通のセクター・バイトの構成例を示した図である。

【図24】図24は、シグナリング・メッセージの1つであるデータ・トランスミッション・メッセージの構成例を示した図である。

10

【図25】図25は、データ・アセット・マネジメント・テーブル (DAMT) 2500の構成例を示した図である。

【図26】図26は、データ・ロケーション・マネジメント・テーブル (DLMT) 2600の構成例を示した図である。

【図27】図27は、データ・コンテンツ・マネジメント・テーブル (DCMT) 2700の構成例を示した図である。

【図28】図28は、MMT伝送されるデータ放送アプリケーション (コンテンツ) の伝送、ロケーションと提示を行なう仕組みを説明するための図である。

【図29】図29は、MMT伝送路からデータ放送アプリケーション (コンテンツ) を取得する際の、シグナリング情報として伝送される各テーブルの参照関係を説明するための図である。

20

【図30】図30は、受信機内でアプリケーションを事前キャッシュする仕組みを模式的に示した図である。

【図31】図31は、受信機において放送ストリームからアプリケーションを取得して起動する動作を説明するための図である。

【図32】図32は、受信機においてアプリケーション提示画面が遷移する際の動作を説明するための図である。

【図33】図33は、アプリケーションのファイル・データが更新された際の受信機における動作を説明するための図である。

30

【発明を実施するための形態】

【0028】

以下、図面を参照しながら本明細書で開示する技術の実施形態について詳細に説明する。

【0029】

図1には、本明細書で開示する技術を適用したデジタル放送システム10の構成例を模式的に示している。図示のデジタル放送システム10は、放送送出システム11と、受信機12で構成される。

【0030】

放送送出システム11は、伝送メディアを含むIP (Internet Protocol) 方式の放送信号を送信する。放送信号の伝送メディアには、タイムド・メディアと、ファイルのようなノンタイムド・メディアの両方が含まれる。タイムド・メディアは、例えば、ビデオやオーディオ、字幕などのストリーム・データである。また、ノンタイムド・メディアは、例えばHTML (Hyper Text Markup Language) 文書のようなアプリケーション (コンテンツ) のファイル・データである。

40

【0031】

一方、受信機12は、放送送出システム11から送られてくる放送信号を受信する。そして、受信機12は、受信した放送信号からビデオやオーディオ、字幕などの伝送メディアを取得して、画像や音声を提示する。

【0032】

50

図1に示したデジタル放送システム10では、放送送出システム11から受信機12へ放送信号を伝送する際のトランスポート方式として、MMTを適用することを想定している。図2には、この場合の放送信号構成例をスタック・モデル200で示している。

【0033】

スタック・モデル200の最下層には、物理レイヤー(PHY)201がある。物理例201には、変調方式や誤り訂正方式などが含まれる。

【0034】

物理レイヤー201の上に、TLV(Type Length Value)の伝送パケットのレイヤー202がある。また、TLV202の上にはIPパケット203が載り、さらにその上にUDP(User Datagram Protocol)204が載る。また、TLVの伝送パケット202の上には、IP203とUDP204のヘッダーを圧縮したヘッダー圧縮IP205と、シグナリング(Signaling)情報としての伝送制御信号206も載る。

【0035】

UDP204の上には、MMTパケット207、現在時刻の情報を含むNTP(Network Time Protocol)パケット208などが載る。MMTプロトコル(MMTP)は、MMTPペイロード209をIPネットワーク上で伝送するためのアプリケーション・レイヤーのトランスポート・プロトコルである。

【0036】

MMTパケット207のMMTPペイロード209には、MFU(MMT Fragment Unit)210あるいはシグナリング・メッセージ(Signaling Message)211が含まれる。MFU210は、符号化されたタイムド・メディア並びにノンタイムド・メディアのコンテナであるMPU(Media Processing Unit)のフラグメントである。MFU210には、ビデオやオーディオ、字幕などのストリーム・データ(タイムド・メディア)212や、HTML文書データなどのファイル・データ(ノンタイムド・メディア)213が挿入される。

【0037】

図3には、図2に示した放送信号を送出する放送送出システム11の構成例を示している。図示の放送送出システム11は、時計部301と、信号送出部302と、ビデオ・エンコーダー303と、オーディオ・エンコーダー304と、キャプション・エンコーダー305と、シグナリング・エンコーダー306と、ファイル・エンコーダー307と、情報システム308と、TLVシグナリング・エンコーダー309と、IPサービス・マルチプレクサー(MUX)310と、TLVマルチプレクサー(MUX)311と、変調・送信部312を備えている。

【0038】

時計部301は、NTPサーバー(図示しない)から取得した時刻情報に同期した時刻情報を生成し、この時刻情報を含むIPパケットをIPサービス・マルチプレクサー310に送る。

【0039】

信号送出部302は、例えばTV放送局のスタジオやVTRなどの記録再生機であり、タイムド・メディアであるビデオ、オーディオ、字幕などのストリーム・データや、ノンタイムド・メディアであるHTML文書データなどのファイル・データをそれぞれ、ビデオ・エンコーダー303、オーディオ・エンコーダー304、キャプション・エンコーダー305、ファイル・エンコーダー307に送る。また、情報システム308は、TV放送局のスケジューラー並びにファイルの供給源であり、ノンタイムド・メディアであるHTML文書データ、シグナリング情報をそれぞれ、ファイル・エンコーダー307、シグナリング・エンコーダー306に送る。

【0040】

ビデオ・エンコーダー303は、信号送出部302から送出されるビデオ信号を符号化し、さらにパケット化して、ビデオのMMTパケットを含むIPパケットをIPサービス

10

20

30

40

50

・マルチプレクサー 310 に送る。また、オーディオ・エンコーダー 304 は、信号送出部 302 から送出されるオーディオ信号を符号化し、さらにパケット化して、オーディオの MMT パケットを含む IP パケットを IP サービス・マルチプレクサー 310 に送る。また、キャプション・エンコーダー 305 は、信号送出部 302 から送出される字幕信号を符号化し、さらにパケット化して、字幕の MMT パケットを含む IP パケットを IP サービス・マルチプレクサー 310 に送る。

【0041】

シグナリング・エンコーダー 306 は、情報システム 308 から送出される情報に基づいてシグナリング・メッセージを生成し、ペイロード部にこのシグナリング・メッセージが配置された MMT パケットを含む IP パケットを IP サービス・マルチプレクサー 310 に送る。本実施形態では、生成されるメッセージは、PA メッセージ、M2 セクション・メッセージ、データ・トランスミッション・メッセージの 3 種類に大別されるが、各シグナリング・メッセージの詳細については後述に譲る。

【0042】

ファイル・エンコーダー 307 は、信号送出部 302 又は情報システム 308 から送出されるファイル・データを、必要に応じて分割して、ファイル・データを含む MMT パケットを生成し、この MMT パケットを含む IP パケットを IP サービス・マルチプレクサー 310 に送る。なお、ファイル・データは、データ放送コンテンツ（データ放送用アプリケーション）を構成するものである。

【0043】

放送送出システム 11 は、送出するチャンネル（放送番組）毎に IP サービス・マルチプレクサー 310 を装備する。1 つのチャンネルの IP サービス・マルチプレクサー 310 は、各エンコーダー 303 ~ 307 から送られてくるビデオ、オーディオ、字幕、シグナリング・メッセージ、及びファイル・データの各々を含む IP パケットをマルチプレクスして、1 つのチャンネルを構成する TLV パケットを生成する。

【0044】

TLV シグナリング・エンコーダー 309 は、情報システム 308 から送出されるシグナリング情報をエンコードして、ペイロード部に配置する TLV パケットを生成する。

【0045】

TLV マルチプレクサー 311 は、各 IP サービス・マルチプレクサー 310 - 1 ~ 310 - N 及び TLV シグナリング・エンコーダー 309 で生成される TLV パケットをマルチプレクスして、放送ストリームを生成する。

【0046】

変調・送信部 312 は、TLV マルチプレクサー 311 で生成された放送ストリームに対して RF 変調処理を行なって、RF 伝送路に送出する。

【0047】

図 3 に示した放送送出システム 11 の動作について説明しておく。

【0048】

時計部 301 では、NTP サーバーから取得した時刻情報に同期した時刻情報が生成され、この時刻情報を含む IP パケットが生成される。

【0049】

信号送出部 302 から送出されるビデオ信号は、ビデオ・エンコーダー 303、に供給される。ビデオ・エンコーダー 303 では、ビデオ信号が符号化され、さらにパケット化されて、ビデオの MMT パケットを含む IP パケットが生成される。この IP パケットは、IP サービス・マルチプレクサー 310 に送られる。

【0050】

また、信号送出部 302 から送出されるオーディオ信号、字幕信号に対しても、同様の処理が行なわれる。そして、また、オーディオ・エンコーダー 304 で生成されるオーディオの MMT パケットを含む IP パケットが IP サービス・マルチプレクサー 310 に送られ、キャプション・エンコーダー 305 で生成される字幕の MMT パケットを含む IP

10

20

30

40

50

パケットがIPサービス・マルチプレクサー310に送られる。

【0051】

また、シグナリング・エンコーダー306では、情報システム308から送出される情報に基づいてシグナリング・メッセージを生成され、ペイロード部にこのシグナリング・メッセージが配置されたMMTパケットを含むIPパケットが生成される。このIPパケットは、IPサービス・マルチプレクサー310に送られる。

【0052】

また、信号送出部302又は情報システム308から送出されるファイル・データは、ファイル・エンコーダー307に供給される。ファイル・エンコーダー307では、ファイル・データが必要に応じて分割され、ファイル・データを含むMMTパケットが生成され、このMMTパケットを含むIPパケットが生成される。このIPパケットは、IPサービス・マルチプレクサー310に送られる。

【0053】

各IPサービス・マルチプレクサー310では、各エンコーダー303～307から送られてくるビデオ、オーディオ、字幕、シグナリング・メッセージ、及びファイル・データの各々を含むIPパケットがマルチプレクスされて、1つのチャンネルを構成するTLVパケットが生成される。

【0054】

TLVシグナリング・エンコーダー309では、情報システム308から送出されるシグナリング情報がエンコードされて、ペイロード部に配置するTLVパケットを生成する。

【0055】

TLVマルチプレクサー311では、各IPサービス・マルチプレクサー310-1～310-N及びTLVシグナリング・エンコーダー309で生成されるTLVパケットがマルチプレクスされて、放送ストリームが生成される。変調・送信部312では、TLVマルチプレクサー311で生成された放送ストリームに対してRF変調処理が行われ、そのRF変調信号がRF伝送路に送出される。

【0056】

また、図4には、図2に示した放送信号を受信する受信機12の構成例を示している。図示の受信機12は、チューナー・復調部401と、デマルチプレクサー（DEMUX）402と、時計部403と、ビデオ・デコーダー404と、オーディオ・デコーダー405と、キャプション・デコーダー406と、データ放送アプリケーション・エンジン407と、システム制御部408と、合成部409を備えている。

【0057】

チューナー・復調部401は、RF変調信号を受信して、復調処理を行なって、放送ストリームを得る。デマルチプレクサー402は、この放送ストリームに対して、デマルチプレクス処理及びパケット化処理を行なって、NTP時刻情報、PTS（Presentation Time Stamp：提示時刻情報）、シグナリング情報、ビデオ、オーディオ、キャプションの各符号化信号、ファイル・データ、並びにシグナリング情報を出力する。なお、ファイル・データは、データ放送コンテンツ（データ放送用アプリケーション）を構成するものである。

【0058】

システム制御部408は、デマルチプレクサー402で得られるシグナリング情報や、ユーザー操作部（図示しない）を介したユーザーからの操作情報などに基づいて、当該受信機12の各部の動作を制御する。時計部403は、デマルチプレクサー402で得られるNTP時刻情報に基づいて、この時刻情報に同期した時刻情報を生成する。

【0059】

ビデオ・デコーダー404は、デマルチプレクサー402で得られる符号化ビデオ信号をデコードして、ベースバンドのビデオ信号を得る。また、オーディオ・デコーダー405は、デマルチプレクサー402で得られる符号化オーディオ信号をデコードして、ベー

10

20

30

40

50

スバンドのオーディオ信号を得る。また、キャプション・デコーダー 406 は、デマルチプレクサー 402 で得られる符号化字幕信号をデコードして、字幕の表示信号を得る。

【0060】

データ放送アプリケーション・エンジン 407 は、ファイル・データとして伝送されてくるデータ放送アプリケーションの処理部である。すなわち、データ放送アプリケーション・エンジン 407 は、デマルチプレクサー 402 で得られるファイル・データを処理して、データ放送の表示信号を得る。データ放送アプリケーションは、例えば HTML 形式で記述されるファイル・データ (HTML 文書データ) である。本明細書では、データ放送アプリケーションを、「アプリケーション」又は「コンテンツ」とも呼ぶ。

【0061】

なお、放送ストリームでは、同一コンテンツのファイル・データが繰り返し送られてくる。システム制御部 408 は、デマルチプレクサー 402 におけるフィルタリング動作を制御して、繰り返し送られてくるファイル・データ群の中からデマルチプレクサー 402 において必要なもののみがデータ放送アプリケーション・エンジン 407 で取得されるようにする。

【0062】

本実施形態では、システム制御部 408 は、PA メッセージ、M2 セクション・メッセージ、データ・トランスミッション・メッセージの各々に含まれるシグナリング・テーブルを参照して、データ放送を提示するために必要なアクセス範囲を特定し、データ放送アプリケーション・エンジン 407 でキャッシュ可能なファイル・データを事前に取得するためのフィルタリング動作を制御する。ファイル・データの事前キャッシュの詳細については後述に譲る。

【0063】

また、システム制御部 408 は、各デコーダー 404 ~ 406 におけるデコード・タイミングを PTS に基づいて制御し、ビデオ、オーディオ、字幕の提示タイミングを調整する。合成部 409 は、ベースバンドのビデオ信号に、字幕の表示信号及びデータ放送の表示信号を合成し、映像表示用のビデオ信号を得る。また、オーディオ・デコーダー 405 で得られるベースバンドのオーディオ信号は、音声出力用のオーディオ信号となる。

【0064】

図 4 に示した受信機 12 の動作について説明しておく。

【0065】

チューナー・復調部 401 では、RF 変調信号が受信され、復調処理が行なわれて、放送ストリームが得られる。デマルチプレクサー 402 では、この放送ストリームに対して、デマルチプレクス処理及びデパケット化処理を行なわれ、NTP 時刻情報、PTS、シグナリング情報、ビデオ、オーディオ、キャプションの各符号化信号、並びに、ファイル・データが抽出される。

【0066】

デマルチプレクサー 402 で抽出された NTP 時刻情報は、時計部 403 に送られる。時計部 403 では、NTP 時刻情報に基づいて、この時刻情報に同期した時刻情報が生成される。つまり、時計部 403 では、放送送出システム 11 側の時計部 301 で生成された時刻情報に合った時刻情報が生成される。

【0067】

デマルチプレクサー 402 で抽出された符号化ビデオ信号は、ビデオ・デコーダー 404 に送られてデコードされ、ベースバンドのビデオ信号が得られる。また、デマルチプレクサー 402 で抽出された符号化字幕信号はキャプション・デコーダー 406 に送られてデコードされ、字幕の表示信号が得られる。また、デマルチプレクサー 402 で抽出されたファイル・データはデータ放送アプリケーション・エンジン 407 に送られて処理され、データ放送の表示信号が得られる。なお、システム制御部 408 によってデマルチプレクサー 402 におけるフィルタリング動作が制御されて、必要なファイル・データのみがデマルチプレクサー 402 で取得されるようにする。

10

20

30

40

50

【 0 0 6 8 】

そして、合成部 4 0 9 では、ベースバンドのビデオ信号に、字幕の表示信号及びデータ放送の表示信号が合成され、映像表示用のビデオ信号が得られる。

【 0 0 6 9 】

また、デマルチプレクサー 4 0 2 で抽出された符号化オーディオ信号はオーディオ・デコーダー 4 0 5 に送られてデコードされ、音声出力用のベースバンドのオーディオ信号が得られる。

【 0 0 7 0 】

図 1 に示したデジタル放送システム 1 0 では、放送送出システム 1 1 から受信機 1 2 へ放送信号を伝送する際のトランスポート方式として、MMT を適用することを想定している。図 5 には、MMT 方式に従って放送送出システム 1 1 から RF 伝送路に送出される放送信号 5 0 0 のイメージを示している。

10

【 0 0 7 1 】

1 つのチャンネル（放送番組）の放送信号は、ビデオ、オーディオ、字幕などのタイムド・メディアと、ファイル・データのようなノンタイムド・メディアで構成され、これらをエンコードしたメディア・データを MPU に格納して伝送する。また、これらの放送信号の伝送制御などに関する情報を、シグナリング情報として伝送する。MMT では、1 つのチャンネル（放送番組）を構成するタイムド・メディア及びノンタイムド・メディアのデータを異なる伝送路の組み合わせで利用することが容易である。図 5 に示す例では、放送信号 5 0 0 として、ビデオ、オーディオ、字幕、ファイル・データ、シグナリング情報など、データのタイプ毎の MMT 伝送路 5 0 1 ~ 5 0 4 が利用されている。すなわち、図中、字幕データ用の伝送路は便宜上、図示を省略している。

20

【 0 0 7 2 】

1 つのチャンネル（放送番組）は、ビデオ、オーディオ、字幕、ファイル・データ（アプリケーション）などタイプの異なる複数のアセットで構成される「パッケージ」と言うことができる（パッケージは、MMT 伝送路を使値って伝送されるメディア・データの論理集合である）。各アセットは、同じ `asset_id` を共有する 1 又はそれ以上の MPU の集合（論理グループ）であり、それぞれ専用の ES（Elementary Stream）すなわち MMT 伝送路上で伝送される（アセットは、固有の識別子に関連付けられ、マルチメディアのプレゼンテーションを構成するために使用されるデータのエンティティである）。すなわち、伝送路 5 0 1 では、共通の `asset_id` を持つ MPU 論理グループからなるビデオの MMT パケット（MMTP）が伝送され、伝送路 5 0 2 では、共通の `asset_id` を持つ MPU 論理グループからなるオーディオの MMT パケットが伝送され、伝送路 5 0 3 では、共通の `asset_id` を持つ MPU 論理グループからなるファイル・データの MMT パケットが伝送される。MPU は、`asset_id` と、該当する伝送路上での MPU のシーケンス番号で特定される。また、各メディアを伝送する MMT 伝送路は、`asset_id` で識別することができる。

30

【 0 0 7 3 】

付言すれば、1 つのパッケージ（放送番組）で、タイプが同じ複数の（すなわち、`asset_id` が異なる）アセットが伝送されることもある。例えば、同じ放送番組に対して、2 以上のファイル・コンテンツ（データ放送アプリケーション）が提供される場合である。このような場合、異なるファイル・コンテンツには別々の `asset_id` が割り振られ、別々の MPU 論理グループとして異なる MMT 伝送路上で伝送されることになる。図 5 では、簡素化のため、ファイル・データ用の伝送路 5 0 3 を 1 本しか描いていない。

40

【 0 0 7 4 】

また、MMT は、放送や通信の複数の伝送路に共通に用いることができる。HTML 文書データのようなノンタイムド・メディアは、図 5 に示したように放送の伝送路でタイムド・メディアとともに伝送される以外に、通信の伝送路を介して提供することもできる。

【 0 0 7 5 】

50

また、伝送路504では、同じシグナリング・メッセージを含んだMMTパケットが、繰り返し伝送される。本明細書で開示する技術を実現する上で、伝送されるシグナリング・メッセージは、PAメッセージ510、M2セクション・メッセージ520、データ・トランスミッション・メッセージ530の3種類のシグナリング・メッセージが関連する。各種シグナリング・メッセージで、シグナリング・テーブルが伝送される。例えば、PAメッセージ510内には、MP(MMT Package)テーブル511が含まれている。また、M2セクション・メッセージ520内には、MH AI(Application Information)テーブル521が含まれている。また、データ・トランスミッション・メッセージ530は、データの伝送方法やデータ管理の制御方法を通知するためのメッセージであり、データ・ロケーション・マネジメント・テーブル531、データ・アセット・マネジメント・テーブル532、データ・コンテンツ・マネジメント・テーブル533の各シグナリング・テーブルが含まれている。各テーブルの詳細については後述に譲る。

10

【0076】

上述したように、MMTPでは、ビデオ、オーディオ、字幕などのタイムド・メディアや、ファイル・データのようなノンタイムド・メディアが伝送される。図6には、MMTPパケット600の構成例を示している。MMTPパケットは、MMTプロトコルを用いて伝送されるようにフォーマットされたメディア・データのユニットである。詳細については、例えば非特許文献1を参照されたい。

【0077】

参照番号601で示すパケット・カウンター・フラグ「C」に1が代入されていると、参照番号602で示すパケット・カウンターのフィールドがこのMMTPパケット内に存在することが表される。パケット・カウンター602は、MMTPパケットをカウントした整数値を書き込む32ビット長のフィールドであり、MMTPパケットを送信する度に1ずつインクリメントされる。

20

【0078】

参照番号603で示す拡張フラグ「X」に1が代入されていると、参照番号604で示す拡張ヘッダー604が存在することが表される。図6の下には、拡張ヘッダー604の構成例を併せて示している。拡張ヘッダー604は、参照番号604-1で示す16ビット長のtypeフィールドと、参照番号604-2で示すlengthフィールドと、参照番号604-3で示すheader__extensin__valueフィールドで構成される。lengthフィールドには、header__extensin__valueフィールドのバイト長が書き込まれる。header__extensin__valueフィールドには、MMTの仕様から外れた拡張情報を書き込むことができる。

30

【0079】

参照番号606で示すtypeフィールドには、当該MMTPパケットのペイロード・データのタイプを表すタイプ値が書き込まれる。タイプ値の定義を以下の表1に示しておく。

【0080】

【表 1】

Value	Data type	Definition of data unit
0x00	MPU	a media-aware fragment of the MPU
0x01	Generic object	A generic object such as a complete MPU or an object of another type
0x02	Signaling message	one or more signaling messages or a fragment of a signaling message
0x03	Repair symbol	a single complete repair symbol

10

【 0 0 8 1 】

参照番号 6 0 5 で示す R A P (R a n d o m A c c e s s P o i n t) フラグに 1 が代入されていると、当該 M M T P パケットのペイロードが当該データ・タイプのデータ・ストリームへの R a n d o m A c c e s s P o i n t を含んでいることを表す。

【 0 0 8 2 】

参照番号 6 0 7 で示す、16ビット長の `packet_id` フィールドには、アセットを区別するための整数値が書き込まれる。このフィールドの値は、当該 M M T P パケットが属するアセットの `asset_id` に由来する。`packet_id` と `asset_id` のマッピングは、シグナリング・メッセージの一部である M M T パッケージ (M P) テーブルで示されている。

20

【 0 0 8 3 】

参照番号 6 0 8 で示す、32ビット長の `timestamp` フィールドには、当該 M M T P パケットの送信時間が、N T P プロトコルで規定されている `short-format` で記載される。

【 0 0 8 4 】

参照番号 6 0 9 で示す、32ビット長の `packet_sequence_number` フィールドには、同一の `packet_id` を持つパケットを識別するための整数値 (M M T 伝送路上でのシーケンス番号) が記載される。

30

【 0 0 8 5 】

図 7 には、ノンタイムド・メディアを伝送する M M T P パケットの場合の拡張ヘッダー 7 0 0 の構成例を示している。図示のように、この場合、`length` フィールド 7 0 1 には、`header_extensin_value` フィールドのバイト長として 4 が書き込まれる。`header_extensin_value` フィールドには、`download_id` が 4 バイトで記載される。

【 0 0 8 6 】

M M T プロトコルを使って M P U を伝送する際、送信側及び受信側ではそれぞれパケット化、デパケット化が必要である。パケット化により、M P U は M M T P ペイロードに挿入され、M M T P パケットで伝送される。M M T P ペイロードのフォーマットは、大きなペイロードの伝送が可能ないように、M M T P ペイロードのフラグメンテーションを許容する。また、M T P ペイロードのフォーマットは、小さなデータ・ユニットに対応して、複数の M M T P ペイロードを単一の M M T P ペイロードに挿入するアグリゲーションも許容する。受信側では、デパケット化して、元の M P U データを復元する。

40

【 0 0 8 7 】

図 8 には、M P U モードの場合の M M T P ペイロード 8 0 0 の構成例を示している。詳細については、例えば非特許文献 1 を参照されたい。M P U モードは、M M T P ヘッダーの `type` フィールド 6 0 6 に「0 x 0 0」が書き込まれている場合である。M P U モードの M M T P パケットは、ビデオ、オーディオ、ファイル・データ (アプリケーション)

50

の伝送に使用される。

【0088】

参照番号801で示すMPU Fragment Type (FT)フィールドには、フラグメントのタイプが4ビットの値で示される。FT値の定義を以下の表2に示しておく。

【0089】

【表2】

FT	Description	Content
0	MPU metadata	contains the ftyp, mmpu, moov, and meta boxes as well as any other boxes that appear in between.
1	Movie fragment metadata	contains the moof box and the mdat box, excluding all media data inside the mdat box.
2	MFU	contains a sample or sub-sample of timed media data or an item of non-timed media data.

10

【0090】

参照番号802で示すTimed (T)フラグに1が記入されているときには、タイムド・メディアを伝送するMPUがフラグメントされていることを示し、0が記入されているときには、ノンタイムド・メディアを伝送するMPUがフラグメントされていることを示す。

【0091】

参照番号803で示すFragmentation Identifier (f_i)フィールドは、ペイロード内のデータ・ユニットのフラグメンテーションに関する情報を、2ビットで表す。f_iの4つの値の定義を以下の表3に示しておく。

【0092】

【表3】

Value	Definition of data unit
00	ペイロードは整数個のデータ・ユニットを含む。
01	ペイロードは最初のフラグメントのデータ・ユニットを含む
10	ペイロードは中間のフラグメントのデータ・ユニットを含む
11	ペイロードは最後のフラグメントのデータ・ユニットを含む

30

40

【0093】

当該ペイロードが複数のデータ・ユニットをアグリゲートしたものであるときには、参照番号804で示すaggregation (A)フラグに1が記入される。

【0094】

参照番号805で示す、8ビット長のfragment_counterフィールドには、当該MMTPペイロードが続く同じデータ・ユニットのフラグメントを含んでいるペイロードの数が記載される。

【0095】

50

参照番号 806 で示す、16ビット長の `DU__length` フィールドには、当該フィールドに続くデータ (`DU: Data Unit`) の長さが記載される。但し、Aフラグ 804 が 0 のときは、`DU__length` フィールド 806 はない。

【0096】

参照番号 807 で示す `DU__Header` は、データ・ユニットのヘッダーである。但し、FT値 801 が 0 又は 1 のとき（言い換えれば、MFUでないとき）には、`DU__Header` 807 はない。MFUは、タイムド・メディアのサンプル若しくはサブサンプル、又は、ノンタイムド・メディアのアイテムを含んでいる。

【0097】

図 9 には、ペイロードにタイムド・メディアを配置した MFU の `DU__Header` 900 の構成例を示している。また、図 10 には、ペイロードにノンタイムド・メディアを配置した MFU の `DU__Header` 1000 の構成例を示している。図 10 に示すように、ノンタイムド・メディアの場合の `DU__Header` 1000 は、当該 MFU の一部として伝送されるアイテムの識別子である 32 ビット長の `item__ID` で構成される。アイテムは、HTML 文書データや、HTML 文書から参照されるモノメディア・データなどの、アプリケーションを構成するリソースである。`asset__id` で指定された MMT 伝送路上では、上述した MMT P パケットのヘッダー内の `packet__id` 及び拡張ヘッダー内の `download__id` と、DU ヘッダー内の `item__ID` の組み合わせで、アイテムを一意に特定することができる。

【0098】

図 11 には、ノンタイムド・メディアのデータを伝送する際のパケット構成例を示している。

【0099】

図 11 (a) には、元のファイル・データの状態を示している。同図中、F1、F2 はそれぞれ 1 つのファイル・データである。ファイル・データは、例えば HTML 文書であり、1 以上のアイテムを含んでいる。また、HTML 文書自体も 1 つのアイテムである。

【0100】

図 11 (b) には、各ファイル・データ F1、F2 を MFU に配置した様子を示している。ファイル・データ F1 は、ファイル・サイズが大きいので、そのまま 1 つの MFU のペイロードに配置される。一方、ファイル・データ F2 は、ファイル・サイズが大きいので、複数個に分割され、それぞれが MFU のペイロードに配置される。図示の例では、ファイル・データ F2 は、F2 - 1 と F2 - 2 に 2 分割され、それぞれが別の MFU のペイロードに配置されている。

【0101】

ここで、HTML 文書データやモノメディアなどのノンタイムド・メディアがペイロードに配置される MFU には、そのアイテムを一意に示す `item__ID` が記載された `DU__Header` (図 10 を参照のこと) がそれぞれ付けられる。

【0102】

次いで、図 11 (c) に示すように、各 MFU には、MMT ペイロードのヘッダー (図 8 を参照のこと) が付けられて、MMT ペイロードとなる。ここで、MMT ペイロードのヘッダーの `Fragment Type (FT)` フィールドに値 2 を記載して、フラグメントのタイプが MFU であることを示す。また、`Timed (T)` フラグに値 0 を記載して、ノンタイムド・メディアを伝送する MPU であることを示す。また、フラグメントしていないノンタイムド・メディアを配置した MFU には、`Fragmentation Identifier (f__i)` フィールドに値 0 を記載する。一方、フラグメントしたノンタイムド・メディアを配置した MFU には、`Fragmentation Identifier (f__i)` フィールドに値 1 を記載するとともに、`fragment__counter` フィールドに該当するカウント値を記載する。

【0103】

次いで、図 11 (d) に示すように、各 MMT ペイロードに、MMT P パケットのヘッ

10

20

30

40

50

ダー及び拡張ヘッダー（図6を参照のこと）が付けられて、MMTパケット・ストリームとなる。ここで、MMTPヘッダーのtypeフィールドには、0を記載して、ペイロード・データのタイプがMPUであることを記載し、packet_idフィールドにはアセットを区別するための整数値が書き込まれる。また、拡張ヘッダーには、download_idが記載される。したがって、asset_idで指定されたMMT伝送路上では、上述したMMTPパケットのヘッダー内のpacket_id及び拡張ヘッダー内のdownload_idと、DUヘッダー内のitem_IDの組み合わせで、アイテムを一意に特定することができる。

【0104】

さらに、図11(e)に示すように、各MMTパケットにIPヘッダー及びUDPヘッダーが付けられて、IPパケット・ストリームとなる。図示を省略したが、各IPパケットにTLVヘッダーを付けることで、放送ストリームを構成するTLVパケットが生成される。

10

【0105】

なお、図11では図示を省略したが、MMTパケットには、シグナリング・メッセージをペイロードに含むMMTパケットも存在する。シグナリング・メッセージは、PAメッセージ、M2セクション・メッセージ、データ・トランスミッション・メッセージがある（前述並びに図5を参照のこと）。MMTPペイロードにタイムド・メディアやノンタイムド・メディアなどの伝送メディアが含まれるか、あるいは、シグナリング・メッセージが含まれるかは、MMTPヘッダー内のtypeフィールドの値を参照して識別することができる。

20

【0106】

続いて、本明細書で開示する技術を実現する上で関連する、MMTプロトコルで使用されるシグナリング・メッセージの構成について説明する。シグナリング・メッセージは、パッケージの伝送制御やパッケージの使用に必要なシグナリング情報であり、各種のシグナリング・テーブルを伝送する。

【0107】

MMTのシグナリング・メッセージは、3つの共通するフィールドと、シグナリング・メッセージ・タイプ毎の特定の1つのフィールドと、メッセージ・ペイロードからなる一般的なフォーマットを使用する。メッセージ・ペイロードは、シグナリング情報を伝送する。以下、PAメッセージ、M2セクション・メッセージ、データ・トランスミッション・メッセージの順に説明する。

30

【0108】

PA(Packege Access)メッセージは、Packege Accessに必要なすべてのシグナリング・テーブル上の情報を持つPAテーブルを伝送する。PAテーブルには、MMT Package(MP)テーブルが含まれる。図12には、シグナリング・メッセージの1つであるPAメッセージ1201と、PAメッセージに含まれるMPテーブル1202の構成例を示している。また、図13には、PAメッセージ1300のシンタックス例を示し、図14には、PAメッセージに含まれるパラメータの説明を示している。

40

【0109】

message_idは、各種シグナリング情報において、PAメッセージを識別する16ビットの固定値である。versionは、PAメッセージのバージョンを示す、8ビットの整数値のパラメータである。例えばMPテーブルを構成する一部のパラメータでも更新した場合には、versionは+1だけインクリメントされる。lengthは、このフィールドの直後からカウントされる、当該PAメッセージのサイズをバイト単位で示す、32ビット長のパラメータである。

【0110】

extensionフィールドには、payloadのフィールドに配置されるMPテーブル(MPT)のインデックス情報が配置される。このフィールドには、8ビットのt

50

`table__id`と、8ビットの`table__version`と、16ビットの`table__length`が配置される。`table__id`は、MPテーブルを識別する固定値である。`table__version`は、MPテーブルのバージョンを示す。`table__length`は、MPテーブルのサイズをバイト単位で示す。

【0111】

PAメッセージの`payload`フィールドには、MPテーブルが配置される。MPテーブルは、すべてのアセットのリストを含むパッケージに関連する情報を格納する。

【0112】

図15及び図16には、MPテーブルのシンタックス例を示している(図16は、図15の続く後半部分である)。また、図17には、MPテーブルに含まれるパラメータの説明を示している。以下、MPテーブルの構成について説明する。

10

【0113】

`table__id`は、各種シグナリング情報においてMPテーブルであることを識別する8ビットの固定値である。`version`は、MPテーブルのバージョンを示す8ビットの整数値である。例えば、MPテーブルを構成する一部のパラメータでも更新した場合には、`version`は+1だけインクリメントされる。`length`は、このフィールドの直後からカウントされる、MPテーブルのサイズをバイト単位で示す、32ビット長のパラメータである。

【0114】

`MMT__package__id`は、放送信号で伝送されるすべての信号(ビデオ、オーディオ、字幕)、並びにファイルなどのアセットを構成要素とする全体のパッケージとしての識別情報である。この識別情報は、テキスト情報である。`MMT__package__id__length`は、そのテキスト情報のサイズをバイト単位で示す。

20

【0115】

`MP__table__descriptors`のフィールドは、パッケージ全体に関わる記述子の格納領域である。`MP__table__descriptor__length`は、そのフィールドのサイズ $N2$ をバイト単位で示す、16ビット長のパラメータである。そして、`MP__table__descriptor`は、さまざまな目的の記述子を規定した上で、 $N2$ バイト分(1つ又は複数配置)することを想定している。

【0116】

`number__of__assets`は、パッケージを構成する要素としてのアセット(信号、ファイル)の数を示す、8ビットのパラメータである。`number__of__asset`の数分($N3$)だけ、以下の`Asset__loop`が配置される。

【0117】

1つの`Asset__loop`内には、個々のアセットの情報としての`asset__id__len`、`asset__id`、`gen__loc__info`、`asset__dsc__len`、`asset__descriptor`の各パラメータが配置される。

【0118】

`asset__id`は、アセットをユニークに識別するテキスト情報である。`asset__id__len`は、`asset__id`のサイズをバイト単位で示す。`gen__loc__info`は、アセットの取得先のロケーションを示す情報である。本実施形態では、`gen__loc__info`は、アセットの取得先となる伝送路上の`packet__id`の形式で記述される。したがって、MPテーブル上で`asset__id`を引いて、MMT伝送路上の該当する`packet__ID`を取り出すことができる。

40

【0119】

`asset__descriptor`のフィールドは、アセットに関わる記述子の格納領域である。`asset__descriptors__length`は、`asset__descriptor`フィールドのサイズ $N5$ をバイト単位で示す。そして、`asset__descriptor`は、さまざまな目的の記述子を規定した上で、 $N5$ 個(1つ又は複数)配置することを想定している。

50

【0120】

M2セクション・メッセージは、MPEG-2 Systemのセクション拡張形式をそのまま伝送するために用いるシグナリング・メッセージである。図18には、M2セクション・メッセージ1800の構成例を示している。以下、M2セクション・メッセージの各パラメーターの意味について説明する。

【0121】

message_id (メッセージ識別) は、各種シグナリング情報において、M2セクション・メッセージを識別する16ビットの固定値であり、本実施形態では0x8000とする。version (バージョン) は、M2セクション・メッセージのバージョンを示す、8ビットの整数値のパラメーターである。length (メッセージ長) は、このフィールドの直後からカウントされる、当該M2セクション・メッセージのサイズをバイト単位で示す、16ビット長のパラメーターである。table_id (テーブル識別) は、当該セクションが属するテーブルの識別のために使用する領域である。section_syntax_indicator (セクション・シンタックス指示) は、拡張形式を示す「1」とする。section_length (セクション長) は、セクション長領域より後に続くデータのバイト数を書き込む領域である。table_id_extension (テーブル識別拡張) は、テーブル識別の拡張を行なう領域である。version_number (バージョン番号) は、テーブルのバージョン番号を書き込む領域である。current_next_indicator (カレント・ネクスト指示) は、テーブルが現在使用可能である場合は「1」とし、テーブルが現在使用不可であり次に有効となることを示す場合は「0」とする。section_number (セクション番号) は、テーブルを構成するセクション番号を書き込む領域である。last_section_number (最終セクション番号) は、テーブルを構成する最後のセクション番号を書き込む領域である。CRC32 (CRC)、ITU-T勧告H.222.0に従う巡回冗長符号とする。

【0122】

図19には、M2セクション・メッセージで伝送されるMH AI (Application Information) テーブル (MH AIT) 1900の構成例を示している。以下、MH AI テーブルの各パラメーターの意味について説明する。

【0123】

table_id (テーブル識別) は、各種シグナリング情報においてアプリケーション情報 (AI) テーブルであることを識別する8ビットの固定値であり、本実施形態では0x89とする。section_syntax_indicator (セクション・シンタックス指示) は、1ビットのフィールドで、常に「1」とする。section_length (セクション長) は、12ビットのフィールドで、先頭の2ビットは常に「00」とする。これは、セクション長フィールドからCRC32を含むセクションの最後までセクションのバイト数を規定する。この値は1021 (16進数で0x3FD) を超えないものとする。application_type (アプリケーション形式) は、16ビットのフィールドで、AITで伝送しているアプリケーションの値を示す。DVBでは、DVB-Jアプリケーションに対して0x0001が割り当てられている。ARIB-Jアプリケーションにおいても0x0001とする。version_number (バージョン番号) は、5ビットのフィールドで、サブテーブルのバージョン番号である。version_number は、当該MH AI テーブルのバージョン番号であり、サブテーブル内の情報に変化があった場合に+1だけインクリメントされる。また、バージョン番号の値が「31」になったとき、その次は「0」に戻る。current_next_indicator (カレント・ネクスト指示) は、常に「1」とする。section_number (セクション番号) は、8ビットのフィールドで、セクションの番号を表す。サブテーブル内で最初のセクションのセクション番号は0x00である。セクション番号は、同一のテーブル識別及びアプリケーション形式を持つセクションが追加される度に+1だけインクリメントされる。last_section_number (最終セク

10

20

30

40

50

ション番号)は、8ビットのフィールドであり、そのセクションが属するサブテーブルにおける最後のセクション番号を規定する。

【0124】

`common_descriptor_length` (共通記述子ループ長)は、8ビットのフィールドで、後続の`descriptor` (記述領域内記述子)のバイト長を規定する。この`descriptor` (記述領域内記述子)は、`common_descriptor_length`の数分のループからなる一連の領域に、記述子(`descriptor`)の情報を格納する。`descriptor`は、AITサブテーブル内のすべてのアプリケーションに適用される。例えば、伝送プロトコル記述子が`descriptor`フィールドに書き込まれる。

10

【0125】

`application_loop_length`は、このMH AIテーブルに含まれるアプリケーション情報の数を書き込む領域である。そして、`application_loop_length`が示す数分だけ、アプリケーション情報のループが配置される。

【0126】

1つのアプリケーション情報のループ内には、`application_identifier` (アプリケーション識別子)と、`application_control_code` (アプリケーション制御コード)と、`application_descriptor_loop_length` (アプリケーション情報記述子ループ長)の数分のループからなる一連の領域に記載される`descriptor` (アプリケーション情報記述子)が配置される。この記述子領域内の記述子は、指定したアプリケーションのみに適用される。

20

【0127】

`application_identifier` (アプリケーション識別子)は、アプリケーションを識別するパラメータである。`application_control_code` (アプリケーション制御コード)は、8ビットのフィールドで、アプリケーションの状態を制御する制御コードを規定する。このフィールドのセマンティックスは、アプリケーション形式の値に依存する。`application_control_code`として“`autostart`”が指示されていたら、このMH ATテーブルを参照した受信機は、`application_identifier`で指定されたアプリケーションを起動開始する。また、`application_control_code`として“`prefetch`”が指示されていたら、このMH ATテーブルを参照した受信機は、`application_identifier`で指定されたアプリケーションを先読みする。また、`application_control_code`として“`kill`”が指示されていたら、このMH ATテーブルを参照した受信機は、`application_identifier`で指定されたアプリケーションの実行を停止する。CRC32 (CRC)、ITU-T勧告H.222.0に従う巡回冗長符号とする。

30

【0128】

要するに、MH AIテーブルは、MMT伝送路で送られてくるアプリケーション(ファイル・データ)の処理方法や、伝送方法(`transport_protocol`)、ロケーション(URL)を指定するテーブルである。受信機は、M2セクション・メッセージで送られてくるMH AIテーブルを受信すると、`application_control_code`で指定された処理を実行するために、指定されたロケーションから指定された`transport_protocol`でアプリケーションを取得する。

40

【0129】

図20には、MH AIテーブルのアプリケーション情報のループ内に格納される、アプリケーション情報記述子2000の構成例を示している。また、図21には、アプリケーション情報記述子2000に含まれるパラメータの説明を示している。以下、アプリケーション情報記述子2000の各パラメータの意味について説明する。

50

【0130】

`descriptor_tag`は、当該記述子2000を識別する、8ビットの整数値である。`descriptor_length`は、このフィールドより後に続く当該記述子2000のデータのバイト数を書き込む領域である。

【0131】

`application_profile_length`の数分のループからなる一連の領域には、`application_profile`の情報が書き込まれる。`application_profile`は、本アプリケーションが実行可能である受信機のプロファイルである。受信機に要求する機能毎のビットマップで要求機能を示す。但し上位3ビットは機能ビットマップ切り替えを示す。上記ビットマップはバージョン毎に規定する。また、`version_major`、`version_minor`、`version_micro`はそれぞれ、アプリケーション・プロファイル規定のバージョンである。

10

【0132】

`service_bound_flag`は、本アプリケーションが現在のサービスのみで有効かどうかを示すフラグである。`visibility`は、アプリケーション可視か否かを示す。`application_priority`は、このサービス内で告知されているアプリケーション間の相対優先度である。`transport_protocol_label`は、アプリケーションを伝送するプロトコルを示す。`transport_protocol_label`の値としては、0x0003はHTTP/HTTPS伝送、0x0005はMMT並びにノンタイムド伝送を規定する。

20

【0133】

また、図22には、伝送プロトコル記述子2200の構成例を示している。以下、伝送プロトコル記述子2200の各パラメーターの意味について説明する。

【0134】

`descriptor_tag`は、当該記述子2200を識別する、8ビットの整数値である。`descriptor_length`は、このフィールドより後に続く当該記述子2200のデータのバイト数を書き込む、8ビットの領域である。

【0135】

`protocol_id`(プロトコルID)は、アプリケーションを伝送するプロトコルを示す。値としては、0x0003はHTTP/HTTPS伝送、0x0005はMMT並びにノンタイムド伝送を規定する。`transport_protocol_label`(伝送プロトコル・ラベル)は、1つのアプリケーションを複数の経路で伝送する場合にその伝送手段を一意に識別する値であり、アプリケーション情報記述子の同名のフィールドに対応する。`selector_byte`(セレクター・バイト)は、プロトコルID毎にシンタックスが規定される領域であり、取得場所が書き込まれる。

30

【0136】

図23には、HTTP/HTTPS、MMTノンタイムド伝送に共通のセレクター・バイト2300の構成例を示している。

【0137】

`url_base_byte`は、`url_base_length`の数分のループからなる一連の領域に、URL文字列のうち、`url_base`を示すテキスト情報を格納する。

40

【0138】

`url_extension_count`は、`url_base`に続くURL extensionの数を示し、`url_extension_count`の数分だけURL extensionのループが配置される。そして、1つのURL extensionのループ内では、`url_extention_byte`は、URL extensionの長さを規定するURL extension_lengthの数分のループからなる一連の領域に、個々のURL extentionを示すテキスト情報を格納する。各URL extentionは、`url_base`に続くURL文字列である。例えば、U

50

R L _ b a s e が “ h t t p : / / w w w . x b c . c o m ” で、U R L _ e x t e n s i o n が “ i n d e x . h t m l ” であれば、これらの文字列を連結して、完全な URL “ h t t p : / / x b c . c o m / i n d e x . h t m l ” を得ることができる。

【 0 1 3 9 】

要するに、M H A I テーブルのアプリケーション情報のループ内のアプリケーション情報記述子並びに伝送プロトコル記述子を参照することで、アプリケーションの伝送手段（MMT 伝送か、HTML 伝送か）、並びに、ロケーション情報（URL）を取得することができる。

【 0 1 4 0 】

図 2 4 には、シグナリング・メッセージの 1 つであるデータ・トランスミッション・メッセージ 2 4 0 0 の構成例を示している。以下、データ・トランスミッション・メッセージの各パラメーターの意味について説明する。

【 0 1 4 1 】

m e s s a g e _ i d（メッセージ識別）は、各種シグナリング情報において、データ・トランスミッション・メッセージを識別する 1 6 ビットの固定値であり、本実施形態では 0 x F 0 0 0 とする。v e r s i o n（バージョン）は、データ・トランスミッション・メッセージのバージョン番号を書き込む領域である。l e n g t h（メッセージ長）は、このフィールドより後に続く当該メッセージのデータのサイズをバイト単位で示す、3 2 ビットのパラメーターである。

【 0 1 4 2 】

n u m _ o f _ t a b l e s（テーブル数）は、このデータ・トランスミッション・メッセージに格納するテーブルの数を示す。データ・トランスミッション・メッセージに格納するテーブルとして、そして、n u m _ o f _ t a b l e s が示す数分だけ、テーブル情報のループが配置される。

【 0 1 4 3 】

1 つのテーブル情報のループ内には、テーブル情報として、t a b l e _ i d（テーブル識別）、t a b l e _ v e r s i o n（テーブル・バージョン）、並びに、t a b l e _ l e n g t h（テーブル長）が格納される。t a b l e _ i d（テーブル識別）は、このデータ・トランスミッション・メッセージに格納するテーブルの識別のための使用する領域である。データ・トランスミッション・メッセージで、データ・ロケーション・マネジメント・テーブル、データ・アセット・マネジメント・テーブル、データ・コンテンツ・マネジメント・テーブルの 3 種類のシグナリング・テーブルが伝送されるが（前述）、t a b l e _ i d はこれらのうちいずれのテーブルであるかを識別する。t a b l e _ v e r s i o n（テーブル・バージョン）は、このデータ・トランスミッション・メッセージに格納するテーブルのバージョンを示す。t a b l e _ l e n g t h（テーブル長）は、このデータ・トランスミッション・メッセージに格納するテーブルの大きさをバイト単位で示す。t a b l e（テーブル）は、このデータ・トランスミッション・メッセージに格納するテーブルを示す。

【 0 1 4 4 】

また、n u m _ o f _ t a b l e s が示す数分だけ、テーブルのループが配置される。1 つのテーブルのループ内には、t a b l e _ i d で識別されるテーブルの中身の情報が格納される。

【 0 1 4 5 】

図 2 5 には、データ・トランスミッション・メッセージで伝送されるデータ・アセット・マネジメント・テーブル 2 5 0 0 の構成例を示している。データ・アセット・マネジメント・テーブルは、MMTP パケットとして伝送されるファイル・データのアセットの情報と、ファイル・データの各アセットに含まれるアイテムの情報を管理するテーブルである。以下、このデータ・アセット・マネジメント・テーブルの各パラメーターの意味について説明する。

【 0 1 4 6 】

10

20

30

40

50

`table_id` (テーブル識別) は、各種シグナリング情報においてデータ・アセット・マネジメント・テーブルであることを示す 8 ビットの固定値であり、本実施形態では `0xA2` とする。`version_` (バージョン) は、このデータ・アセット・マネジメント・テーブルのバージョンを示す 8 ビットの整数値のパラメータである。例えばデータ・アセット・マネジメント・テーブルを構成する一部のパラメータでも更新した場合には、`version` は +1 だけインクリメントされる。`length` は、このフィールドの直後からカウントされる、このデータ・アセット・マネジメント・テーブルのサイズをバイト単位で示す、16 ビット長のパラメータである。

【0147】

`number_of_asset` は、パッケージに含まれるファイル・データのアセットの数を示す、8 ビットのパラメータである。`number_of_asset` の数分だけ、以下の `Asset loop` が配置され、アセット毎のファイル・データの情報が格納される。

【0148】

1 つの `Asset loop` 内には、`download_id` と、アセット (ファイル・データ) 自体に関する情報と、そのアセットに含まれる各アイテムに関する情報が含まれる。`download_id` は、ノンタイムド・メディア (ファイル・データ) を伝送する MMT P パケットの拡張ヘッダーに書き込まれる識別情報である (図 7 を参照のこと)。

【0149】

`Asset loop` 内に格納されるアセット自体に関する情報として、`asset_ID_scheme`、`asset_ID_length`、`asset_ID_length` と、`asset_ID_byte` を含む。`asset_ID_scheme` は、アセット ID の形式を示す。アセット ID の形式として、例えば `UUID (Universal Unique Identifier)`、`URI (Uniform Resource Identifier)`、`GURL (General URL)` を割り当てることができる。`asset_ID_length` は、アセット ID バイトの長さをバイト単位で表す。`asset_ID_byte` は、`asset_ID_length` の数分のループからなる一連の領域に、`asset_ID_scheme` で指定された形式で、アセット ID を示す。ちなみに、この情報は、本実施形態では、MP テーブル、データ・アセットマネジメント・テーブル共通にアセットを識別する情報として用いられるが、データ量が大きいので他の代用可能なアセット識別情報を用いてもよい。例えば、MP テーブルにおいて `asset_ID` に対応する情報として 16 ビットの `component_tag` を定義し、データ・アセット・マネジメント・テーブルにおいては `asset_ID` の代わりに `component_tag` を利用することが想定される。

【0150】

`number_of_items` は、該当するファイル・データのアセットを構成するアイテムの数を書き込む領域である。そして、`number_of_items` の数分だけアイテムのループが配置され、アセット (ファイル・データ) を構成する各アイテムに関する情報が書き込まれる。

【0151】

1 つのアイテムのループ内には、アイテムに関する情報として、`item_ID`、`item_tag`、`item_size`、`item_version`、`item_checksum`、`item_info` の各パラメータが記述される。`item_ID` は、ノンタイムド MFU で伝送されるアイテムを識別する ID を示す 32 ビットの値である。`item_tag` は、同様にアイテムを識別する情報であり、16 ビットの値である。シグナリング情報としては、32 ビットの `item_ID` に代えて 16 ビットの `item_tag` を使用することで、アイテムの識別に必要なビット・サイズを削減することができる。`item_size` は、アイテムのサイズをバイト単位で表す。`item_version` は、アイテムのバージョンを示し、アイテムの内容が更新される度に `version` は +

10

20

30

40

50

1だけインクリメントされる。`item_checksum`は、アイテムのチェックサムを示す。なお、チェックサムは、すべてのファイルに対して必ず設定するのは情報量が多いと考えられる。よって、そのような考慮により、例えば1ビットの`checksum_flag`を設定し、これに1が代入された場合にのみ32ビットの`item_checksum`が現れるようにしてもよい。あるいは、シグナリングではなく、図7に示したMMTPパケットの拡張ヘッダーとして`type`としてチェックサムを示し、`length`の後に32ビットのチェックサムを配置してもよい。`item_info_length`は、`item_info_byte`の情報領域のサイズをバイト単位で表す。そして、`item_info_byte`は、`item_info_length`の数分のループからなる一連の領域に、当該アイテムに関する情報を格納する。

10

【0152】

`descriptor_loop_length`は、`descriptor`の全バイト長を示す。`descriptor`は、`descriptor_loop_length`の数分のループからなる一連の領域に記述子(`descriptor`)の情報を格納する。格納される記述子は別途定義する。

【0153】

要するに、データ・アセット・マネジメント・テーブル2500は、1つのパッケージに含まれるファイル・データ(コンテンツ)のアセット並びにアセットに含まれるアイテムに関する情報を管理するテーブルである。アイテムに関する情報として、アイテムのバージョン情報も管理する。データ・アセット・マネジメント・テーブル2500を参照して、`item_tag`(若しくは、`Item_ID`)から該当する`asset_id`やアセットを伝送するMMTP拡張ヘッダーに記載された`downloadID`や`item_info`を引いたり、シグナリング情報の伝送路上で扱う`item_tag`からファイル・データの伝送路上の`item_ID`や`item_info`を引いたりすることができる。

20

【0154】

図26には、データ・トランスミッション・メッセージで伝送されるデータ・ロケーション・マネジメント・テーブル(DLMT)2600の構成例を示している。データ・ロケーション・マネジメント・テーブルは、MMTPパケットとして伝送される各ファイル・データのロケーション情報と、ファイル・データに含まれる各アイテムのロケーション情報を管理するテーブルである。以下、このデータ・ロケーション・マネジメント・テーブルの各パラメーターの意味について説明する。

30

【0155】

`table_id`(テーブル識別)には、各種シグナリング情報においてデータ・ロケーション・マネジメント・テーブルであることを示す8ビットの固定値が書き込まれる。`version`(バージョン)は、このデータ・ロケーション・マネジメント・テーブルのバージョンを示す8ビットの整数値のパラメーターである。例えばデータ・ロケーション・マネジメント・テーブルを構成する一部のパラメーターでも更新した場合には、`version`は+1だけインクリメントされる。`length`は、このフィールドの直後からカウントされる、このデータ・ロケーション・マネジメント・テーブルのサイズをバイト単位で示す、16ビット長のパラメーターである。

40

【0156】

`base_URL_length`は、`base_URL_byte`の情報領域のサイズをバイト単位で表す。`base_URL_byte`は、`base_URL_length`の数分のループからなる一連の領域に、ファイル・データの絶対的なURL形式で表記されたロケーション情報を格納する。

【0157】

`number_of_items`は、ファイル・データに含まれるアイテムの数を書き込む領域である。そして、`number_of_items`の数分だけアイテムのループが配置される。

【0158】

50

1つのアイテムのループ内には、ファイル・データに含まれる各アイテムについての、`item_tag`、`item_URI_byte`が書き込まれる。`item_tag`は、ノンタイムドMFUで伝送されるアイテムを識別する情報を、32ビットの`item_ID`(前述)よりも短い16ビットで表す。`item_URI_byte`は、`number_of_item_URI_byte`の数分のループからなる一連の領域に、ファイル・データのベースとなるロケーション情報すなわち`base_URL_byte`に対する相対的なURL形式で表記されたロケーション情報を格納する。例えば、コンテンツの`base_URL`が“`http://www.xbc.com`”で、あるアイテムの`item_URL`が“`index.html`”であれば、これらの文字列を連結して、完全なURL“`http://www.xbc.com/index.html`”を得ることができる。

10

【0159】

要するに、データ・ロケーション・マネジメント・テーブル2600は、1つのパッケージに含まれるファイル・データ(コンテンツ)並びにファイル・データに含まれるアイテムに関するロケーション情報を管理するテーブルである。データ・ロケーション・マネジメント・テーブル2600を参照して、`item_tag`からそのアイテムのURLを引いたり、逆にURLから該当する`item_tag`を引いたりすることができる。なお、`base_URL`で示されるロケーション(ディレクトリー)の下にさらに複雑なディレクトリー構造を設定したい場合には、本構成例では`item_URI_byte`の情報量が大きくなり過ぎる可能性がある。これを考慮すると、データ・ロケーション・マネジメント・テーブルにおいて、ファイルが存在するディレクトリーのロケーション情報を

20

【0160】

また、図27には、データ・トランスミッション・メッセージで伝送されるデータ・コンテンツ・マネジメント・テーブル(DCMT)2700の構成例を示している。データ・コンテンツ・マネジメント・テーブルは、ノンタイムド・メディアとして伝送されるファイル・データすなわちコンテンツ(データ放送アプリケーション)の情報を管理するテーブルである。以下、このデータ・コンテンツ・マネジメント・テーブルの各パラメーターの意味について説明する。

30

【0161】

`table_id`(テーブル識別)には、各種シグナリング情報においてデータ・コンテンツ・マネジメント・テーブルであることを示す8ビットの固定値が書き込まれる。`version`(バージョン)は、このデータ・コンテンツ・マネジメント・テーブルのバージョンを示す8ビットの整数値のパラメーターである。例えばデータ・コンテンツ・マネジメント・テーブルを構成する一部のパラメーターでも更新した場合には、`version`は+1だけインクリメントされる。`length`は、このフィールドの直後からカウントされる、このデータ・コンテンツ・マネジメント・テーブルのサイズをバイト単位で示す、16ビット長のパラメーターである。

【0162】

40

`number_of_content`は、パッケージに含まれるコンテンツの数を示す、8ビットのパラメーターである(コンテンツは、例えば、データ放送アプリケーションを記述したHTML文書などのファイル・データである)。`number_of_content`の数分だけ、以下のコンテンツのループが配置され、コンテンツ毎の情報が格納される。

【0163】

1つのコンテンツのループ内には、コンテンツに関する情報として、`content_ID`と、`content_version`と、`content_cache_size`と、当該コンテンツに含まれるアプリケーション提示単位(Presentation Unit:PU)に関する情報が書き込まれる。`content_ID`は、コンテンツの識

50

別情報である。 `content__version` は、コンテンツのバージョンを示す。 `content__cache__size` は、コンテンツをキャッシュするサイズを示す。

【0164】

`number__of__PU` は、コンテンツに含まれるアプリケーション提示単位 `PU` の数であり、 `number__of__PU` の数分だけ `PU` のループが配置される。

【0165】

1つの `PU` のループ内には、 `PU` の識別情報である `PU__tag` と、 `PU` をキャッシュするサイズを示す `PU__cache__size` が書き込まれる。また、 `PU` のループ内には、該当する `PU` に含まれるアイテムの数を示す `number__of__items` と、 `number__of__items` の数分だけのアイテムのループが配置される。1つのアイテムのループ内には、アイテムの `item__tag` が書き込まれる。

10

【0166】

また、1つの `PU` のループ内には、この `PU` からリンクされる他の `PU` の数を示す `number__of__linked__PU` と、 `number__of__linked__PU` の数分だけの `linked__PU` のループが配置される。1つの `linked__PU` のループ内では、 `linked__PU` の識別情報である `linked__PU__tag` が書き込まれる。

【0167】

要するに、データ・コンテンツ・マネジメント・テーブル2700は、1つのパッケージで各コンテンツ（データ放送アプリケーション）をアプリケーション提示単位（`PU`）で管理するテーブルである。データ・コンテンツ・マネジメント・テーブル2700を参照して、 `item__tag` から、そのアイテムを含むアプリケーション提示単位 `PU__tag` を取得することができる。なお、本構成例では、データ・コンテンツ、`PU`、アイテムという階層構造としたが、キャッシュを利用したアプリケーション以外の一般的なデータ・コンテンツを想定した場合には、データ・コンテンツ、アイテムという2階層にした上で、アイテム毎の情報として `PU__tag` を指定できるようにしてもよい。また、データ・コンテンツ・マネジメント・テーブルを利用しないで、同様の情報をデータ・アセット・マネジメント・テーブルに設定する方法も考えられる。その場合、図25に示すデータ・アセット・マネジメント・テーブルの `item__info()` に配置可能な記述子により同等の情報を表現する。具体的には、例えばデータ・マネジメント記述子としてアイテムが属するべき1つ又は複数の `content__ID` 及び `PU__tag` を指定できるようにする。

20

30

【0168】

図28には、MMT伝送されるデータ放送アプリケーション（コンテンツ）の伝送、コンテンツのロケーションと、アプリケーションの提示を行なう仕組みを図解している。

【0169】

図28(A)には、コンテンツのディレクトリ構造を示している。各コンテンツ `content 1`、`2`、... は、アプリケーション（`app`）とマテリアルで構成される。アプリケーションやマテリアルは、それぞれファイル・データが実体であるリソースである。各リソースは、MMT伝送路上ではアセットの構成要素であるアイテムに相当し、 `item__ID` で識別することができる。図28(C)に示すように、各リソースは、該当するアセットのMMT伝送路上でアイテムとして伝送される（後述）。アプリケーションは、コンテンツの実行時（アプリケーションの提示時）において参照される1以上のHTML文書からなる。また、マテリアルは、HTML文書から参照される `jpeg` 画像やその他のタイプのモノメディア・データなどである。1つのHTML文書と、そこから参照されるマテリアルで、1つのアプリケーション提示単位 `PU` を構成する。図28(A)に示す例では、 `content 1` は、 `A11.html`、 `A12.html`、 `A13.html` などの1以上のHTML文書をアプリケーションのリソースとして持つ。このうち、 `A11.html` は、コンテンツの実行時に直接参照されるリソースとする。

40

【0170】

50

図28(B)には、コンテンツの実行時(アプリケーションの提示時)におけるリソース間の参照関係を示している。図示の例では、コンテンツの実行時に直接参照されるアプリケーションA11とこれが参照するマテリアルB11、B02が1つのアプリケーション提示単位PUを構成するリソース・グループ2801であり、PU__tagとしてp1が割り当てられている(なお、B14は、放送によりMMT伝送されるのではなく通信によるHTTP伝送で随時取得することができるマテリアルであり、以下では、アプリケーション提示単位のリソース・グループには含まないものとして扱う)。

【0171】

同様に、アプリケーションA12とこれが参照するマテリアルB12、B02、B13が1つのアプリケーション提示単位PUを構成するリソース・グループ2802であり、PU__tagとしてp2が割り当てられている(なお、B07は、放送によりMMT伝送されるのではなく通信によるHTTP伝送で随時取得することができるマテリアルであり、以下では、アプリケーション提示単位のリソース・グループには含まないものとして扱う)。同様に、アプリケーションA01とこれが参照するマテリアルB03、B01、B04が1つのアプリケーション提示単位PUを構成するリソース・グループ2803であり、PU__tagとしてp3が割り当てられている。

【0172】

また、複数のHTML文書間でリンク参照関係を持つことができる(周知)。図28(B)に示す例では、リソースA01.htmlは、コンテンツの実行時に直接参照され、最初に表示されるアプリケーション提示画面を記述するHTML文書である。これに対し、同じcontent1に含まれリソースA12.htmlと、content1外のcommonに含まれるリソースA01.htmlは、A01.htmlを実行して提示される画面から遷移するアプリケーション提示画面を記述するHTML文書であり、A11.htmlとリンク参照関係を持つ。各リソースA01.html、A12.html、A01.htmlは、それぞれ1つのアプリケーション提示単位PUを構成するリソース・グループ2801、2802、2803を形成する。そして、リンクし合うアプリケーション提示単位2801、2802、2803同士で、さらに上位の大きなリソース・グループ2810を構成する。

【0173】

また、パッケージ(1つの放送番組)に含まれるアプリケーション全体となるさらにコンテンツ全体で大きなリソース・グループすなわちデータ・コンテンツ全体を構成する。データ・コンテンツ全体とは、共通のcontent_IDを持つアプリケーション提示単位PUの範囲であり、データ・コンテンツ・マネジメントテーブルで、該当するcontent_IDのPUのループを回すことにより、コンテンツに含まれるすべてのアプリケーション提示単位PUを一括して特定することができる。図28(B)に示す例では、content1とcommonに含まれるアプリケーションでコンテンツ全体のリソース・グループ2820を形成している。

【0174】

図28(C)には、コンテンツをMMT伝送する様子を模式的に示している。コンテンツの構成要素であるアプリケーションやマテリアルは、それぞれファイル・データが実体であり、「リソース」とも呼ぶ。各リソースは、MMT伝送路上ではアセットの構成要素であるアイテムに相当する。MMT伝送では、パッケージに含まれる各コンテンツは1つのアセットとして扱われ、それぞれAsset_IDが割り当てられる。図示の例では、content1にはasset_IDとしてa1が割り当てられている。また、MMT伝送では、HTML文書データやマテリアルなどの個々のリソースは、1つのアイテムとして扱われ、それぞれItem_IDが割り当てられる。図示の例では、content1に含まれる各リソースには、それぞれItem_IDとしてi11、i12、i13、i14が割り当てられている。

【0175】

また、同じコンテンツに含まれるリソースは同じasset_IDを共有し、同じMM

10

20

30

40

50

T 伝送路上で伝送される。図 28 (C) に示す例では、Item_ID が i 1 1、i 1 2、i 1 3、i 1 4 の各アイテムは、同じ Asset_ID として a 1 を共有しており、同じ MMT 伝送路上で伝送される。前述したデータ・ロケーション・マネジメント・テーブルは図 28 (A) で表現され、データ・コンテンツ・マネジメント・テーブルは図 28 (B) で表現され、データ・アセット・マネジメント・テーブルは図 28 (C) で表現され、これらの間を item_ID により関係付けられることになる。

【0176】

MMT 伝送路からデータ放送アプリケーション(コンテンツ)を取得する際の、シグナリング情報として伝送される各テーブルの参照関係について、図 29 を参照しながら説明する。

10

【0177】

受信機は、M2 セクション・メッセージで、MH-AI テーブル(MH-AIT) 2901 を取得すると、application_control_code を参照して、アプリケーションの状態がどのように制御されているかを確認する。そして、“autostart” が指示されている場合には、テーブル内の transport_protocol_label を参照して、MMT 伝送が指定されていることを確認すると、このアプリケーションの提示時に直接参照されるアイテム(ファイル・データ)の URL 情報を伝送プロトコル記述子から取り出す。そして、受信機は、データ・トランスミッション・メッセージで送られてくるデータ・ロケーション・マネジメント・テーブル(DLMT) 2902 を参照して、その base_URL 及び item_URL の組み合わせに対応するアイテムの Item_tag を取得することができる。

20

【0178】

次いで、受信機は、データ・トランスミッション・メッセージで送られてくるデータ・アセット・マネジメント・テーブル(DAMT) 2903 を参照して、取得した item_tag を MMT 伝送路上の item_ID に戻すとともに、対応するアセットを特定して、その asset_ID と download_id を取得する。

【0179】

そして、受信機は、PA メッセージで送られてくる MP テーブル(MPT) 2904 を参照して、取得した asset_ID に対応する packet_id を取得すると、ファイル・データの MMT 伝送路上で、MMTP パケットのヘッダー内の packet_id と、拡張ヘッダー内の download_id と、DU ヘッダー内の item_ID に基づいてフィルタリングして、所望する(アプリケーションの提示時に直接参照する)アイテムを取得することができる。

30

【0180】

また、受信機は、データ・トランスミッション・メッセージで送られてくるデータ・コンテンツ・マネジメント・テーブル(DCMT) 2905 内で、データ・ロケーション・マネジメント・テーブル 2902 から取得した item_tag を引いて、該当するアプリケーション提示単位の PU_tag を取り出すことができる。また、この PU_tag の PU のループ内で linked_PU のループを回すことにより、これにリンクする他のアプリケーション提示単位の PU_tag を一括して取り出すことができる。

40

【0181】

トランスポート方式として MMT を採用するデジタル放送システムにおいて、HTML 文書データのようなアプリケーションをノンタイムド・メディアとして伝送する場合、受信機側で、アプリケーション本体並びに参照されているモノメディア・ファイルを放送(MMT 伝送)と通信(HTTP 伝送)の両方の経路で取得することが想定される。

【0182】

受信機において、アプリケーションを実行する場合に迅速な応答を可能とするためには(例えば、リモコンで d ボタンが押されてデータ放送を画面に表示する場合や画面を遷移する場合)、あらかじめ必要なリソース(アイテム、ファイル・データ)を受信し、キャッシュしておくことが好ましい。通信で取得可能なファイル・データはほぼ瞬時に取得す

50

ることができる。一方、放送ストリームではファイル・データは繰り返し送られてくるが、使用可能な帯域が制限されている場合などでは、アプリケーションの実行が指示されてから次にファイル・データが届くまでの時間が長くなり迅速に応答できないおそれがある。このため、放送ストリームで取得するファイル・データに関しては、とりわけ事前にキャッシュしておく必要があると考えられる。

【0183】

そこで、本明細書で開示する技術では、放送送出システム側からは、アプリケーションを構成するリソース(アイテム)を、放送並びに通信の任意のロケーションから取得可能となるように、アプリケーションにおいてロケーションを示すURLを放送などのMMT伝送路上のロケーションにマッピングする情報と、アプリケーションにおける表示単位とリンク関係を示す情報を伝送するようにしている。

10

【0184】

図19～図23を参照しながら説明したように、MH AIテーブルには、アプリケーションの伝送方法(transport_protocol_label)とロケーションを示すURL情報が記載される。一方、図26などを参照しながら説明したように、データ・ロケーション・マネジメント・テーブルには、アプリケーションにおいてロケーションを示すURLとデータ・トランスミッション・メッセージ上のitem_tagとの対応関係が記述され、図25などを参照しながら説明したように、データ・アセット・マネジメント・テーブルには、item_tagと、そのアイテムを含むアセットのasset_ID、そのアセットのMMT伝送路上のMMTPパケットのdownload_id及びitem_IDとの対応関係が記載され、図16などを参照しながら説明したように、MPテーブルでasset_IDを引くことにより、MMT伝送路上でそのアセットを伝送するパケットのpacket_idを取得することができる。したがって、本実施形態では、放送送出システムは、シグナリング・メッセージで伝送されるMH AIテーブル、データ・ロケーション・マネジメント・テーブル、データ・アセット・マネジメント・テーブル、及びMPテーブルを用いて、アプリケーションにおいてロケーションを示すURLを放送などのMMT伝送路上のロケーションにマッピングする情報を伝送することができる。

20

【0185】

また、図27などを参照しながら説明したように、データ・コンテンツ・マネジメント・テーブルには、パッケージに含まれる各コンテンツについて、アプリケーション提示単位の情報を管理する。具体的には、データ・コンテンツ・マネジメント・テーブルは、コンテンツに含まれるアプリケーション提示単位のPU_id、アプリケーション提示単位に含まれるすべてのアイテムのitem_tagと、これにリンクする他のアプリケーション提示単位のPU_id(linked_PU_id)を記載する。また、図25に示したように、データ・アセット・マネジメント・テーブルは、アセットに含まれるすべてのアイテムのサイズ(item_size)をitem_tagと対応付けて管理する。したがって、本実施形態では、放送送出システムは、データ・アセット・マネジメント・テーブルとデータ・コンテンツ・マネジメント・テーブルを用いて、アプリケーションにおける表示単位とリンク関係を示す情報を伝送することができる。

30

40

【0186】

また、本実施形態では、データ・アセット・マネジメント・テーブルでパッケージ内のアプリケーション(ファイル・データ)のアセット並びにアセットに含まれるアイテムに関する情報を管理し、データ・ロケーション・マネジメント・テーブルでパッケージ内のコンテンツ並びにコンテンツに含まれるアイテムに関するロケーション情報を管理し、データ・コンテンツ・マネジメント・テーブルによりアプリケーション提示単位でコンテンツに含まれるアイテム(リソース)を管理する仕組みを採り入れている。

【0187】

したがって、本実施形態では、アプリケーションのオーサリング時におけるディレクトリー構成の自由度と、また、アプリケーションを構成する任意のファイルを放送(MMT

50

伝送)と通信(HTTP伝送)の伝送路に切り分ける自由度と、アプリケーション実行時におけるアプリケーション提示単位でリンク関係の自由度を確保するアプリケーション伝送方式を提供することができる。

【0188】

また、本実施形態では、図28(B)などを参照しながら説明したように、以下の4通りのアクセス範囲(a)~(b)で、コンテンツへのアクセス範囲に関するロケーション情報とサイズを階層的な情報として与えることができる。

【0189】

- (a) アプリケーション実行時に直接参照するリソース(例えば、図28(B)中のA11.html)
- (b) 同時に表示するアプリケーション提示単位を構成するリソース・グループ(例えば、図28(B)中の参照番号2801、2802、2803、2804で示す各リソース・グループ)
- (c) 同時に表示するアプリケーション提示単位を構成するリソース・グループと、これにリンクする他のアプリケーション提示単位を含む大きなリソース・グループ(図28(B)中の参照番号2810で示すリソース・グループ)
- (d) アプリケーション全体のリソース・グループ(図28(B)中の参照番号2820で示すリソース・グループ)

【0190】

したがって、受信機側では、シグナリング・メッセージで伝送される上記の各テーブルに基づいて、キャッシュの空き容量に応じたいずれかのアクセス範囲(a)~(d)と各々のサイズを把握することができ、アプリケーションの効果的な事前キャッシュを行なうことが可能になる。

【0191】

図30には、受信機内で、データ放送アプリケーション・エンジンが処理するアプリケーションを事前キャッシュする仕組みを模式的に示している。

【0192】

図4では図示を省略したが、受信機は、放送信号でMMT伝送されるデータ放送アプリケーションのコンテンツを事前キャッシュするコンテンツ・キャッシュ3001を備えている。あるいは、コンテンツ・キャッシュ3001は、データ放送アプリケーション・エンジン407の内部に配置されていてもよい。

【0193】

システム制御部408は、デマルチプレクサー402で放送ストリームからデマルチプレクスされたシグナリング・メッセージを解析して、受信機内の動作を制御する。コンテンツの事前キャッシュに関しては、システム制御部408は、コンテンツ・キャッシュ3001の空き容量を把握して、可能な限りより大きなアクセス範囲でファイル・データをキャッシュする。

【0194】

具体的には、システム制御部408は、M2セクション・メッセージで伝送されるMHA Iテーブルで指定されたエントリーのアイテム(ファイル・データ、HTML文書データ)のURLを、データ・トランスミッション・メッセージで伝送されるデータ・ロケーション・マネジメント・テーブルで引いて、参照されたアイテムのitem_tagを取得する。データ・アセット・マネジメント・テーブル及びMPテーブルを用いて、item_tagから該当するMMTPパケットを特定できることは既に述べた通りである。

【0195】

次いで、システム制御部408は、データ・トランスミッション・メッセージで伝送されるデータ・コンテンツ・マネジメント・テーブルでitem_tagを引いて、参照されたアイテムが属するアプリケーション提示単位PUとそのサイズ(PU_cache_size)を取得し、さらにそのアプリケーション提示単位とリンクする他のアプリケーション提示単位(linked_PU)並びにそのサイズ、アイテムが属すコンテンツ(

10

20

30

40

50

content_ID)とそのサイズ(content_cache_size)を取得する。すなわち、システム制御部408は、以下の4通りのアクセス範囲(a)~(d)に関するサイズを階層的な情報として把握する。

【0196】

- (a) MH AIテーブルで直接参照したアイテム(例えば、図28(B)中のA11.html)
- (b) 参照したアイテムが属するアプリケーション提示単位(例えば、図28(B)中の参照番号2801)
- (c) 参照したアイテムが属するアプリケーション提示単位と、これにリンクする他のアプリケーション提示単位(例えば、図28(B)中の参照番号2810)
- (d) コンテント(content_IDの範囲)全体(例えば、図28(B)中の参照番号2820)

10

【0197】

そして、システム制御部408は、コンテント・キャッシュ3001の空き容量に基づいて、(a)~(d)のうちいずれのアクセス範囲でアイテムのキャッシュを行なうかを決定する。

【0198】

システム制御部408は、キャッシュしたいアイテムのitem_tagを、データ・トランスミッション・メッセージで伝送されるデータ・アセット・マネジメント・テーブルで引いて、そのアイテムが属するアセットのasset_IDを取得し、次いで、asset_IDをPAメッセージで伝送されるMPテーブルで引いて、アセットが伝送されるMMTPパケットのpacket_idを取得する。また、システム制御部408は、データ・アセット・マネジメント・テーブルから、所望するアイテムを伝送するMMTPパケットの拡張ヘッダーに記載されるdownload_idを取得すると、ファイル・データのMMT伝送路上で、MMTPパケットのヘッダー内のpacket_idと、拡張ヘッダー内のdownload_idと、DUヘッダー内のitem_IDに基づいてフィルタリングして、所望するアイテムのエンティティを取得して、コンテント・キャッシュ3001にキャッシュする。

20

【0199】

また、アクセス範囲(b)、すなわち、参照したアイテムが属するアプリケーション提示単位でキャッシュしたいときには、システム制御部408は、データ・コンテント・マネジメント・テーブルでそのアイテムが属するアプリケーション提示単位PUを特定すると、このPU_tagのPUのループ内でitemのループを回すことにより、同じアプリケーション提示単位PUに含まれるすべてのアイテムのitem_tagを一括して取得する。そして、各アイテムのエンティティを、item_tagを基に上述した手順に従って取得して、逐次コンテント・キャッシュ3001にキャッシュする。

30

【0200】

また、アクセス範囲(c)、すなわち、参照したアイテムが属するアプリケーション提示単位と、これにリンクする他のアプリケーション提示単位でキャッシュしたいときには、システム制御部408は、データ・コンテント・マネジメント・テーブルでそのアイテムが属するアプリケーション提示単位PUのループ内で、linked_PUのループを回すことにより、参照したアイテムが属するアプリケーション提示単位にリンクされたすべてのアプリケーション提示単位PUのPU_tagを一括して取得する。そして、上述した手順に従って、各linked_PUに含まれるに含まれるすべてのアイテムのitem_tagを一括して取得する。そして、各アイテムのエンティティを、item_tagを基に上述した手順に従って取得して、逐次コンテント・キャッシュ3001にキャッシュする。

40

【0201】

また、アクセス範囲(d)、すなわち、コンテント(content_IDの範囲)全体でキャッシュしたいときには、システム制御部408は、システム制御部408は、デ

50

ータ・コンテンツ・マネジメント・テーブルでそのアイテムが属するコンテンツの `content__ID` を特定すると、この `content__id` のループ内で `PU` のループを回すことにより、そのコンテンツに含まれるすべてのアプリケーション提示単位 `PU` の `PU__tag` を一括して取得する。そして、各アイテムのエンティティを、`item__tag` を基に上述した手順に従って取得して、逐次コンテンツ・キャッシュ 3001 にキャッシュする。

【0202】

データ放送アプリケーション・エンジン 407 は、アプリケーションを実行する際、必要なアイテム（ファイル・データ）が既にコンテンツ・キャッシュ 3001 に事前にキャッシュされていれば、デマルチプレクサー 402 で放送ストリームからデマルチプレクスされたファイル・データが届くのを待つことなく、コンテンツ・キャッシュ 3001 から取り出して、迅速に応答して、データ放送用表示信号を生成することができる。

10

【0203】

一方、必要なアイテムがコンテンツ・キャッシュ 3001 内に存在しないときには、データ放送アプリケーション・エンジン 407 は、放送ストリームからデマルチプレクスされたファイル・データが届くのを待って応答して、データ放送用表示信号を生成する。

【0204】

図 31 には、受信機において、放送ストリームからアプリケーションを取得して起動する動作を図解している。

【0205】

システム制御部 408 は、MMT 伝送路 3102 上で受信する各種シグナリング・メッセージの内容を解析する。システム制御部 408 は、M2 セクション・メッセージで伝送される MH AI テーブル 3131 内のアプリケーション情報のループを回して、各アプリケーションの情報を参照する。

20

【0206】

そして、`application__control__code` で “`prefetch`（先読み）” に状態が制御されているアプリケーションを検出すると、システム制御部 408 は、参照番号 3111 で示すように、指定されたエントリーのアイテム（ファイル・データ、HTML 文書データ）の URL を、データ・トランスミッション・メッセージで伝送されるデータ・ロケーション・マネジメント・テーブル 3132 で引いて、参照されたアイテムの `item__tag` を取得する。

30

【0207】

次いで、システム制御部 408 は、参照番号 3112 で示すように、データ・トランスミッション・メッセージで伝送されるデータ・コンテンツ・マネジメント・テーブル 3133 で `item__tag` を引いて、参照されたアイテムが属するアプリケーション提示単位 `PU` を特定すると、その `PU` のループ内のアイテムのループを回して、同じアプリケーション提示単位 `PU` に属する各アイテムの `item__tag` を逐次取得する。

【0208】

次いで、システム制御部 408 は、参照番号 3113 で示すように、各 `item__tag` をデータ・アセット・マネジメント・テーブル 3134 で引いて、そのアイテムが属するアセットの `asset__ID` を取得する。

40

【0209】

次いで、システム制御部 408 は、参照番号 3114 で示すように、`asset__ID` を PA メッセージで伝送される MP テーブル 3135 で引いて、アセットが伝送される MMT P パケットの `packet__id` を取得する。

【0210】

そして、システム制御部 408 は、参照番号 3115 で示すように、データ・アセット・マネジメント・テーブル 3134 で `item__tag` を引いて、MMT 伝送路上の `item__ID` と、所望するアイテムを伝送する MMT P パケットの拡張ヘッダーに記載される `download__id` を取得すると、ファイル・データの MMT 伝送路 3101 上で

50

、MMTPパケットのヘッダー内の`packet_id`と、拡張ヘッダー内の`download_id`と、DUヘッダー内の`item_ID`に基づいてフィルタリングして、参照番号3116で示すように、所望するアプリケーションのアイテムを取得する。取得したアプリケーションのアイテムは、参照番号3117で示すように、逐次コンテンツ・キャッシュ3001に事前キャッシュされる。

【0211】

システム制御部408は、事前キャッシュを行なう際、上述したように、データ放送アプリケーションを実行する際の階層的なファイル・データへのアクセス範囲と、各アクセス範囲におけるファイル・データのロケーション情報及びサイズを取得して、キャッシュの空き容量に応じてアプリケーションを効果的にキャッシュする。

10

【0212】

その後、システム制御部408は、参照番号3118で示すように、受信したMH AIテーブル3136の内容を参照して、`application_control_code`で“`autostart` (自動開始)”に状態が制御されているアプリケーションを検出すると、参照番号3119で示すように、データ放送アプリケーション・エンジン407は、そのアプリケーション識別子(`application_identifier`)で指定されたアプリケーション「A1」を起動する。その際、アプリケーションを構成するファイル・データが事前キャッシュされている場合には、そこからファイル・データをロードして、アプリケーションを迅速に起動することができる。

【0213】

20

また、図32には、受信機において、アプリケーション提示画面が遷移する際の動作を図解している。

【0214】

システム制御部408は、MMT伝送路3202上で受信する各種シグナリング・メッセージの内容を解析する。システム制御部408は、M2セクション・メッセージで伝送されるMH AIテーブル3231内のアプリケーション情報のループを回して、各アプリケーションの情報を参照する。

【0215】

そして、`application_control_code`で“`autostart` (自動開始)”が指示されているアプリケーションを検出すると、システム制御部408は、参照番号3211で示すように、データ放送アプリケーション・エンジン407に対して、そのアプリケーション識別子(`application_identifier`)で指定されたアプリケーション「A1」の起動を指示する。なお、ここでは、アプリケーション「A1」のアイテムが全く事前にキャッシュされていないが、又は、エントリーのアイテムのみが事前にキャッシュされていることを想定する。

30

【0216】

ここで、参照番号3232で示すように、起動したHTMLアプリケーションが、URL“`http://xxx/A2.html`”で指定される他のアイテム(リソース)、すなわちHTML文書データ「A2.html」を参照しているとする。

【0217】

40

このような場合、システム制御部408は、参照番号3212で示すように、そのURLを、データ・トランスミッション・メッセージで伝送されるデータ・ロケーション・マネジメント・テーブル3233で引いて、参照されたアイテムの`item_tag`を取得する。

【0218】

次いで、システム制御部408は、参照番号3213で示すように、データ・トランスミッション・メッセージで伝送されるデータ・アセット・マネジメント・テーブル3234でその`item_tag`を引いて、そのアイテム「A2.html」が属するアセットの`asset_ID`を取得する。

【0219】

50

次いで、システム制御部408は、参照番号3214で示すように、`asset_ID`をPAメッセージで伝送されるMPテーブル3235で引いて、アセットが伝送されるMMTPパケットの`packet_id`を取得する。

【0220】

そして、システム制御部408は、参照番号3215で示すように、データ・アセット・マネジメント・テーブル3234で`item_tag`を引いて、MMT伝送路上の`item_ID`と、所望するアイテムを伝送するMMTPパケットの拡張ヘッダーに記載される`download_id`を取得すると、ファイル・データのMMT伝送路3201上で、MMTPパケットのヘッダー内の`packet_id`と、拡張ヘッダー内の`download_id`と、DUヘッダー内の`item_ID`に基づいてフィルタリングして、参照番号3216で示すように、所望するアイテム「A2.html」を取得する。但し、アイテム「A2.html」が事前にキャッシュされている場合には、MMT伝送路3201から受信する必要はない。そして、データ放送アプリケーション・エンジン407は、アプリケーション「A2.html」を実行し、その結果、提示される文書の画面が遷移する。

10

【0221】

その後、システム制御部408は、参照番号3217で示すように、受信したMH AIテーブル3236の内容を参照して、エントリーのアイテム「A1」について`application_control_code`で“kill(終了)”が指示されていることを検出すると、参照番号3218で示すように、データ放送アプリケーション・エンジン407に対してアプリケーションの終了を指示する。

20

【0222】

また、図33には、受信機において、アプリケーションのファイル・データが更新された際の動作を図解している。

【0223】

システム制御部408は、MMT伝送路3302上で受信する各種シグナリング・メッセージの内容を解析する。システム制御部408は、M2セクション・メッセージで伝送されるMH AIテーブル3331内のアプリケーション情報のループを回して、各アプリケーションの情報を参照する。

【0224】

そして、`application_control_code`で“autostart(自動開始)”が指示されているアプリケーションを検出すると、システム制御部408は、参照番号3311で示すように、データ放送アプリケーション・エンジン407に対して、そのアプリケーション識別子(`application_identifier`)で指定されたアプリケーション「A1」の起動を指示する。

30

【0225】

その後、システム制御部408は、参照番号3312で示すように、データ・トランスミッション・メッセージで伝送されるデータ・アセット・マネジメント・テーブル3332を参照して、上述と同様の手順に従って、参照番号3313で示すように、ファイル・データのMMT伝送路3301上で、MMTPパケットのヘッダー内の`packet_id`と、拡張ヘッダー内の`download_id(=N)`と、DUヘッダー内の`item_ID`に基づいてフィルタリングして、参照番号3313で示すように、該当するMMTPパケット3333を取得する。このMMTPパケット3333で伝送されるアイテムを実行する処理については説明を省略する。

40

【0226】

データ・アセット・マネジメント・テーブル3332では、アイテムのバージョン情報として`item_version=K`を示しており、システム制御部408は、このアイテムのバージョン情報を管理する。

【0227】

システム制御部408は、参照番号3314で示すように、さらにその後に受信したデ

50

ータ・トランスミッション・メッセージのデータ・アセット・マネジメント・テーブル 3334を参照して、同じ `item_id` の `item_version` が K から $K+1$ に更新されていることを検出すると、参照番号 3315 で示すように、データ放送アプリケーション・エンジン 407 に対してファイル更新イベントを通知する。

【0228】

また、システム制御部 408 は、参照番号 3316 で示すように、ファイル・データの MMT 伝送路 3301 上で、MMTP パケットのヘッダー内の `packet_id` と、更新された `download_id (= N+1)` と、DU ヘッダー内の `item_ID` に基づいてフィルタリングして、参照番号 3317 で示すように、バージョンが $K+1$ に更新されたアイテムを伝送する MMT P パケット 3335 を取得する。そして、参照番号 3318 で示すように、取得したアイテムは、アプリケーションのリソースとして、データ放送アプリケーション・エンジン 407 に渡され、文書の画面が遷移する。

10

【0229】

上述したように、本明細書で開示する技術を適用した放送システムでは、アプリケーションを構成するリソース（アイテム）を放送・通信の任意のロケーションから取得可能とするために、アプリケーションにおいてロケーションを示す URL を放送などの MMT 伝送路上のロケーションにマッピングする情報、アプリケーションにおける表示単位とリンク関係を示す情報を伝送することができる。

【0230】

また、本明細書で開示する技術は、アプリケーションのオーサリング時のディレクトリー構成の自由度、アプリケーションを構成する任意のファイルを放送と通信の伝送路に切り分ける自由度、これらとアプリケーション実行時の表示単位、リンク関係の自由度を確保することが可能なアプリケーション伝送方式を提供するものである。

20

【0231】

また、本明細書で開示する技術によれば、放送ストリームの送信側からは、アプリケーションにおいて、直接参照するリソース、同時に表示するアプリケーション提示単位を構成するリソース・グループ、さらにこれにリンクする他のアプリケーション提示単位を含むリソース・グループ、アプリケーション全体のリソース・グループという 4 通りのアクセス範囲で、アクセス範囲に関するロケーション情報とサイズを階層的に情報として与えることができる。これにより、受信側では、利用可能なキャッシュ・サイズに応じて効果的な事前キャッシュを行なうことが可能になる。

30

【産業上の利用可能性】

【0232】

以上、特定の実施形態を参照しながら、本明細書で開示する技術について詳細に説明してきた。しかしながら、本明細書で開示する技術の要旨を逸脱しない範囲で当業者が該実施形態の修正や代用を成し得ることは自明である。

【0233】

本明細書で開示する技術は、トランスポート方式として MMT を採用するさまざまな放送システムに適用することができる。また、本明細書で開示する技術は、タイムド・メディアとノンタイムド・メディアの両方を伝送するさまざまな伝送システムに適用することができる。

40

【0234】

要するに、例示という形態により本明細書で開示する技術について説明してきたのであり、本明細書の記載内容を限定的に解釈するべきではない。本明細書で開示する技術の要旨を判断するためには、特許請求の範囲を参酌すべきである。

【0235】

なお、本明細書の開示の技術は、以下のような構成をとることも可能である。

(1) 所定のトランスポート方式の伝送信号を受信する受信部と、

受信信号を各メディア・データとシグナリング情報にデマルチプレクスするデマルチプレクサーと、

50

デマルチプレクスされたシグナリング情報に基づいて、デマルチプレクスされた少なくとも1つのメディア・データのキャッシュを制御する制御部と、
を具備する受信装置。

(2) 前記所定のトランスポート方式はMMTである、
上記(1)に記載の受信装置。

(3) 伝送信号はタイムド・メディア及びノンタイムド・メディアを含み、
前記制御部は、ノンタイムド・メディアのキャッシュを制御する、
上記(1)に記載の受信装置。

(4) 前記ノンタイムド・メディアは、データ放送アプリケーションのアセットを構成する複数のファイル・データからなり、

前記制御部は、キャッシュの空き容量に応じて、ファイル・データのキャッシュを制御する、
上記(1)に記載の受信装置。

(5) 前記制御部は、受信したシグナリング情報に基づいて、データ放送アプリケーションを実行する際の階層的なファイル・データへのアクセス範囲と、各アクセス範囲におけるファイル・データのロケーション情報及びサイズを取得して、キャッシュを制御する、
上記(4)に記載の受信装置。

(6) 前記制御部は、以下の4通りの階層的なアクセス範囲(a)~(d)とそのサイズを把握して、キャッシュの空き容量に応じていずれのアクセス範囲でキャッシュするかを決定する、

上記(4)又は(5)のいずれかに記載の受信装置。

(a) アプリケーション実行時に直接参照するファイル・データ

(b) 同時に表示するアプリケーション提示単位を構成するファイル・データのリソース・グループ

(c) 同時に表示するアプリケーション提示単位を構成するリソース・グループと、これにリンクする他のアプリケーション提示単位を含む大きなリソース・グループ

(d) アプリケーション全体のリソース・グループ

(7) 前記制御部は、シグナリング情報として、アプリケーションにおいてロケーションを示すURLを放送などのMMT伝送路上のロケーションにマッピングする第1の情報と、アプリケーションにおける表示単位とリンク関係を示す第2の情報を受信して、前記4通りの階層的なアクセス範囲(a)~(d)とそのサイズを把握する、

上記(6)に記載の受信装置。

(8) 前記受信部は、前記第1の情報として、アプリケーションの伝送方法とロケーションを示すURL情報を管理するアプリケーション情報テーブルと、アプリケーションにおいてロケーションを示すURLとデータ・トランスミッション・メッセージ上のitem識別情報との対応関係を管理するデータ・ロケーション・マネジメント・テーブルと、item情報と、そのアイテムを含むアセットのasset識別情報、そのアセットのMMT伝送路上のMMTPパケットのdownload_id及びitem識別情報との対応関係を管理するデータ・アセット・マネジメント・テーブルと、asset識別情報とMMT伝送路上でそのアセットを伝送するパケットのpacket識別情報の対応関係を管理するMMTパッケージ・テーブルを受信する、

上記(7)に記載の受信装置。

(9) 前記受信部は、前記第2の情報として、コンテンツに含まれるアプリケーション提示単位の識別情報、アプリケーション提示単位に含まれるすべてのアイテムのitem識別情報と、これにリンクする他のアプリケーション提示単位の識別情報を管理するデータ・コンテンツ・マネジメント・テーブルを受信する、

上記(7)又は(8)のいずれかに記載の受信装置。

(10) 前記制御部は、prefetch(先読み)に状態が制御されたアプリケーションに関するシグナリング情報を受信したことに応答して、キャッシュ処理を実行する、

上記(1)乃至(9)のいずれかに記載の受信装置。

10

20

30

40

50

(1 1) 所定のトランスポート方式の伝送信号を受信する受信ステップと、
受信信号を各メディア・データとシグナリング情報にデマルチプレクスするデマルチプレクス・ステップと、

デマルチプレクスされたシグナリング情報に基づいて、デマルチプレクスされた少なくとも1つのメディア・データのキャッシュを制御する制御ステップと、
を有する受信方法。

(1 2) タイムド・メディア及びノンタイムド・メディアと、受信側でノンタイムド・メディアのキャッシュ処理に利用可能な情報を含むシグナリング・メッセージをマルチプレクスするマルチプレクサーと、

マルチプレクスされたパケットを所定のトランスポート方式で送信する送信部と、
を具備する送信装置。

10

(1 3) 前記所定のトランスポート方式はMMTである、
上記(1 2)に記載の送信装置。

(1 4) 前記ノンタイムド・メディアは、データ放送アプリケーションのアセットを構成する複数のファイル・データからなり、

前記シグナリング・メッセージは、アプリケーションにおいてロケーションを示すURLを放送などのMMT伝送路上のロケーションにマッピングする第1の情報と、アプリケーションにおける表示単位とリンク関係を示す第2の情報を含む、

上記(1 2)又は(1 3)のいずれかに記載の送信装置。

(1 5) 前記第1の情報は、アプリケーションの伝送方法とロケーションを示すURL情報を管理するアプリケーション情報テーブルと、アプリケーションにおいてロケーションを示すURLとデータ・トランスミッション・メッセージ上のitem識別情報との対応関係を管理するデータ・ロケーション・マネジメント・テーブルと、item識別情報と、そのアイテムを含むアセットのasset識別情報、そのアセットのMMT伝送路上のMMTPパケットのdownload_id及びitem識別情報との対応関係を管理するデータ・アセット・マネジメント・テーブルと、asset識別情報とMMT伝送路上でそのアセットを伝送するパケットのpacket識別除法の対応関係を管理するMMTパッケージ・テーブルを含む、

20

上記(1 4)に記載の送信装置。

(1 6) 前記第2の情報は、コンテンツに含まれるアプリケーション提示単位の識別情報、アプリケーション提示単位に含まれるすべてのアイテムのitem識別情報と、これにリンクする他のアプリケーション提示単位の識別情報を管理するデータ・コンテンツ・マネジメント・テーブルを含む、

30

上記(1 4)又は(1 5)のいずれかに記載の送信装置。

(1 7) タイムド・メディア及びノンタイムド・メディアと、受信側でノンタイムド・メディアのキャッシュ処理に利用可能な情報を含むシグナリング・メッセージをマルチプレクスするマルチプレクス・ステップと、

マルチプレクスされたパケットを所定のトランスポート方式で送信する送信ステップと、
を有する送信方法。

40

【符号の説明】

【0236】

10 ... デジタル放送システム

11 ... 放送送出システム、12 ... 受信機

301 ... 時計部、302 ... 信号送出部、303 ... ビデオ・エンコーダー

304 ... オーディオ・エンコーダー、305 ... キャプション・エンコーダー

306 ... シグナリング・エンコーダー、307 ... ファイル・エンコーダー

308 ... 情報システム、309 ... TLVシグナリング・エンコーダー

310 ... IPサービス・マルチプレクサー

311 ... TLVマルチプレクサー、312 ... 変調・送信部

50

- 4 0 1 ...チューナー・復調部、4 0 2 ...デマルチプレクサー
- 4 0 3 ...時計部、4 0 4 ...ビデオ・デコーダー
- 4 0 5 ...オーディオ・デコーダー、4 0 6 ...キャプション・デコーダー
- 4 0 7 ...データ放送アプリケーション・エンジン
- 4 0 8 ...システム制御部、4 0 9 ...合成部

【要約】 (修正有)

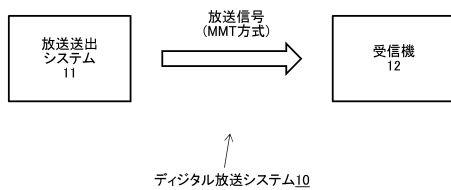
【課題】MMT (MPEG Media Transport) 方式でファイルを受信する受信装置及び受信方法を提供する。

【解決手段】送信側からは、アプリケーションを構成するリソースを、アプリケーションにおいてロケーションを示すURLを放送などのMMT伝送路上のロケーションにマッピングする情報、アプリケーションにおける表示単位とリンク関係を示す情報を伝送する。受信機側では、データ放送アプリケーションを実行する際の階層的なファイル・データへのアクセス範囲と、各アクセス範囲におけるファイル・データのロケーション情報及びサイズを取得して、アプリケーションの効果的な事前キャッシュを行なう。

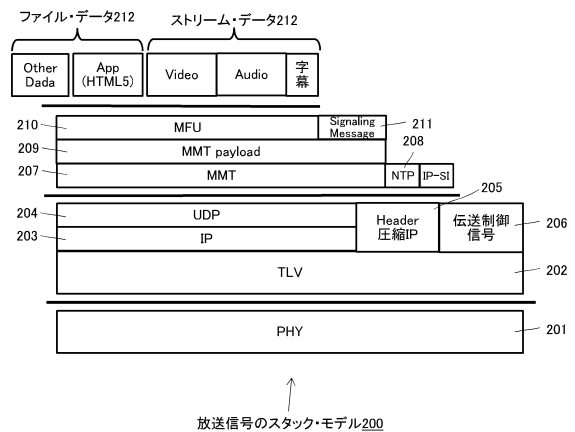
10

【選択図】図 2 8

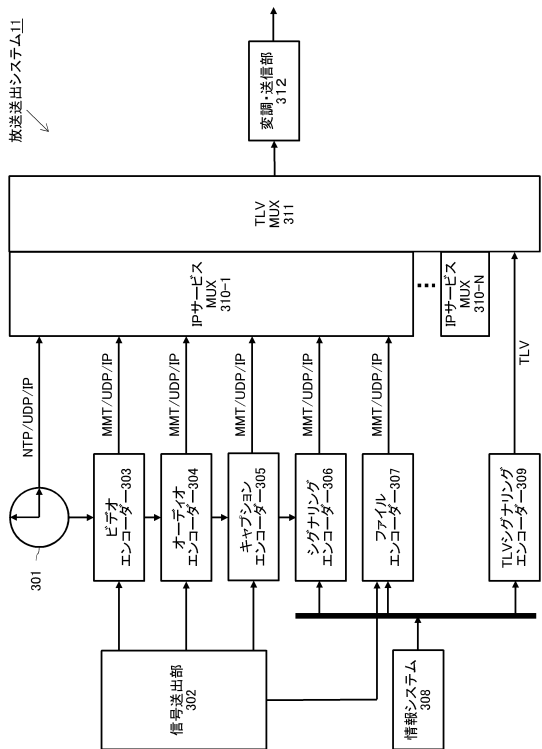
【図 1】



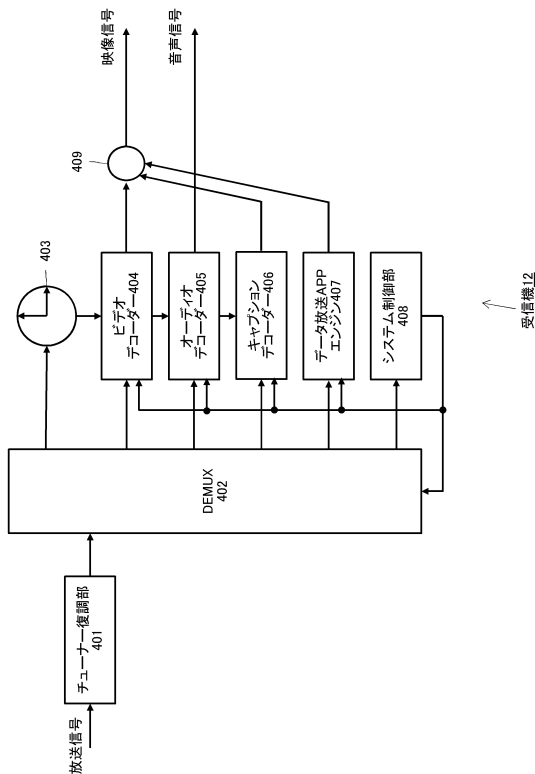
【図 2】



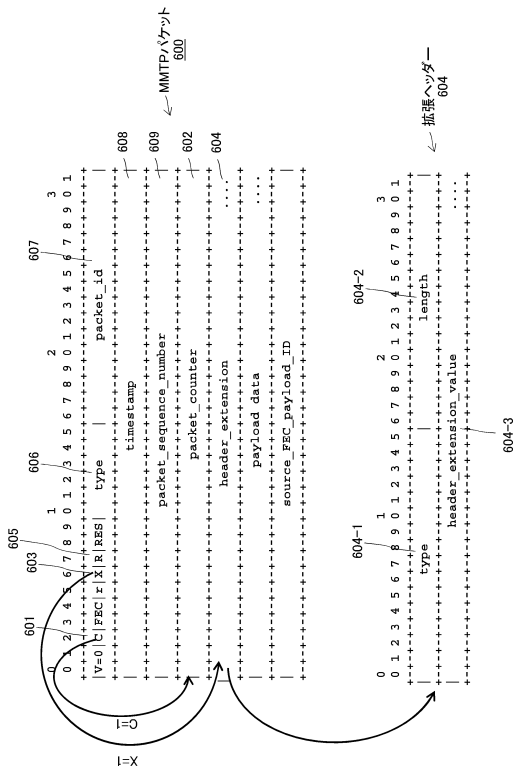
【図3】



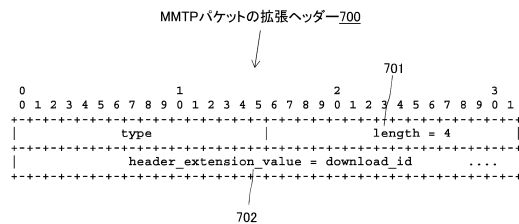
【図4】



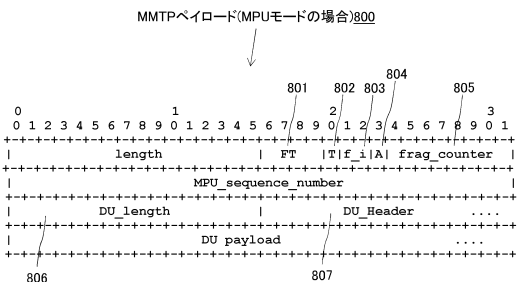
【図6】



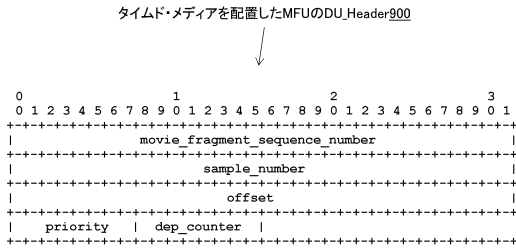
【図7】



【図8】



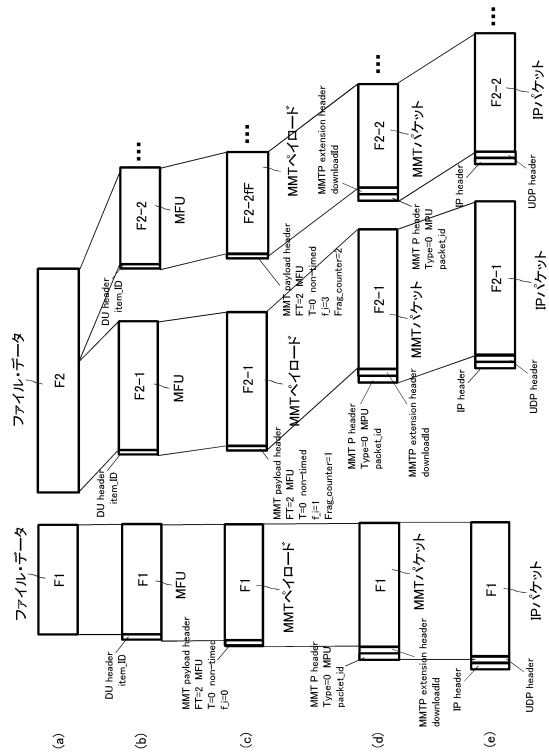
【図9】



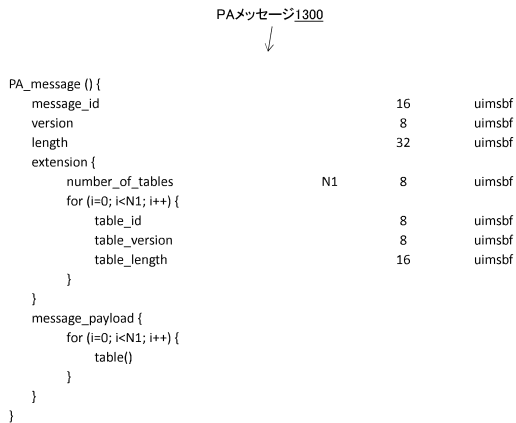
【図10】



【図11】



【図13】



【図14】

項目	日本語項目名	説明
message_id	メッセージID	各種シグナリング情報においてPA messageを識別する固定値
version	バージョン	PA Messageのバージョンを示す。8bit整数値MPTを構成する一部のパラメータでも更新した場合には+1インクリメントされる。
length	テーブル長	PA Messageのバイト数。このフィールドの直後からカウントする。

【 図 1 5 】

MPテーブル(前半部分)

↓

Syntax	Value	No. of bits	Mnemonic
MP_table() {			
table_id		8	uimsbf
version		8	uimsbf
length		16	uimsbf
reserved	'11 1111'	6	bslbf
MP_table_mode		2	bslbf
If (table_id == SUBSET_0_MPT_TABLE_ID) {			
MMT_package_id {	N1		
MMT_package_id_length		8	uimsbf
for (i=0; i<N1; i++) {			
MMT_package_id_byte		8	uimsbf
}			
}			
MP_table_descriptors {	N2		
MP_table_descriptors_length		16	uimsbf
for (i=0; i<N2; i++) {			
MP_table_descriptors_byte		8	uimsbf
}			
}			
}			

【 図 1 6 】

MPテーブル(後半部分)

↓

Syntax	Value	No. of bits	Mnemonic
number_of_assets	N3	8	uimsbf
for (i=0; i<N3; i++) {			
identifier_type		8	uimsbf
asset_id_scheme		32	uimsbf
asset_id_length		8	uimsbf
for(j=0;j<N4;j++){			
asset_id_byte		8	uimsbf
}			
asset_type		32	char
reserved	'1111 111'	7	bslbf
asset_clock_relation_flag		1	bslbf
asset_location {			
location_count	N6	8	uimsbf
for (i=0; i<N6; i++) {			
MMT_general_location_info()			
}			
}			
asset_descriptors {			
asset_descriptors_length	N5	16	uimsbf
for (j=0; j<N5; j++) {			
asset_descriptors_byte		8	uimsbf
}			
}			
}			
}			

【 図 1 7 】

項目	日本語項目名	説明
table_id	テーブルID	各種シグナリング情報においてMP tableを識別する固定値
version	バージョン	MPTのバージョンを示す。8bit整数値MPTを構成する一部のパラメータでも更新した場合には+1インクリメントされる。
length	テーブル長	MP tableのバイト数。このフィールドの直後からカウントする。
package_id	パッケージID	放送信号で伝送される全ての信号、ファイルを構成要素とする全体のパッケージとしての識別情報。
MPT_descriptors	MPT記述子領域	パッケージ全体に関わる記述子の格納領域。記述子はさまざまな目的の記述子を規定した上で1つ又は複数配置する想定。
number_of_assets	アセット数	パッケージを構成する要素としての信号(アセット)の数。この数だけ以下のアセットループが配置される。
asset_id	アセットID	アセットをユニークに識別するID
gen_loc_info	一般ロケーション情報	アセットの取得先のロケーションを示す
asset_descriptors	Asset記述子領域	アセットに関わる記述子の格納領域。記述子はさまざまな目的の記述子を規定した上で1つ又は複数配置する想定。

【 図 1 8 】

M2セクション・メッセージ1800

↓

Syntax	No. of bits	Mnemonic
M2section_message(){		
message_id	16	uimsbf
version	8	uimsbf
length	16	uimsbf
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
'1'	1	bslbf
'11'	2	bslbf
section_length	12	uimsbf
table_id_extension	16	uimsbf
'11'	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for(i=0; i<N; i++){		
signaling_data_byte	8	bslbf
}		
CRC_32	32	rpcof
}		

【 図 19 】

MH AIテーブル(MH AIT)1900

↓

Syntax	No. of bits	Mnemonic
MH-Applicatin_Information_Table(){		
table_id	8	uimbsf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimbsf
application type	16	uimbsf
reserved	2	bslbf
version_number	5	uimbsf
current_next_indicator	1	bslbf
section_number	8	uimbsf
last_section_number	8	uimbsf
reserved_future_use	4	bslbf
common_descriptor_length	8	uimbsf
for(i=0; i<N; i++){		
descriptor ()		rpcof
}		
reserved_future_use	4	uimbsf
application_loop_length		
for(i=0; i<N; i++){		
application_identifier ()		
application_control_code	8	uimbsf
reserved_future_use	4	bslbf
application_descriptor_loop_length	12	uimbsf
for (j=0; j<M; j++) {		
descriptor ()		
}		
}		
}		
CRC32	32	rpcof
}		

【 図 20 】

アプリケーション情報記述子2000

↓

Syntax	No. of bits	Mnemonic
application_descriptor(){		
descriptor_tag	8	uimbsf
descriptor_length	8	uimbsf
application_profile_length	8	uimbsf
for(i=0; i<N; i++){		
application_profile	16	uimbsf
version_major	8	uimbsf
version_minor	8	uimbsf
version_micro	8	uimbsf
}		
service_bound_flag	1	bslbf
visibility	2	bslbf
reserved_future_use	5	bslbf
application_priority	8	uimbsf
for(i=0; i<N; i++){		
transport_protocol_label	8	uimbsf
}		
}		

【 図 21 】

データ	意味
application_profile	本アプリケーションが実行可能である受信機のプロファイル。受信機に要求する機能毎のビットマップで要求機能を示す。但し上位3bitは機能ビットマップ切り替えを示す。上記ビットマップはversion毎に規定する。
version_major version_minor version_micro	アプリケーションプロファイル規定のバージョン。
service_bound_flag	本アプリケーションが現在のサービスのみで有効かどうかを示す。
visibility	アプリケーション可視か否かを示す。
application_priority	このサービス内で告知されているアプリケーション間の相対優先度
transport_protocol_label	アプリケーションを送送するプロトコルを示す。

【 図 22 】

伝送プロトコル記述子2200

↓

Syntax	No. of bits	Mnemonic
transport_protocol_descriptor(){		
descriptor_tag	8	uimbsf
descriptor_length	8	uimbsf
protocol_id	16	uimbsf
transport_protocol_label	8	uimbsf
for(i=0; i<N; i++){		
selector_byte	8	uimbsf
}		
}		

【 図 2 3 】

セレクター・バイト2300

Syntax	No. of bits	Mnemonic
for(i=0;i<N;i++){		
URL_base_length	8	uimsbf
for(j=0;j<N;j++){		
URL_base_byte	8	uimsbf
}		
URL_extension_count	8	uimsbf
for(j=0;j<URL_extension_count;j++){		
URL_extension_length	8	uimsbf
for(k=0;k<URL_extension_length;k++){		
URL_extension_byte	8	uimsbf
}		
}		
}		

【 図 2 4 】

データ・トランсмисシヨん・メッセーヅ2400

Syntax	No. of bits	Mnemonic
Data_Transmission_message(){		
message_id	16	uimsbf
version	8	uimsbf
length	32	uimsbf
num_of_tables	8	uimsbf
for(i=0;i<num_of_tables;i++){		
table_id	8	uimsbf
table_version	8	uimsbf
table_length	16	uimsbf
}		
for(i=0;i<num_of_tables;i++){		
table(i)		
}		
}		

【 図 2 5 】

データ・アセツ・マネヅメント・テーブル(DAMT)2500

Syntax	No. of bits	Mnemonic
Data_Asset_Management_Table(){		
table_id	8	uimsbf
version	8	uimsbf
length	16	uimsbf
number_of_asset	8	Uimsbf
for(k=0;k<number_of_asset;k++){		
download_id	32	uimsbf
asset_ID_scheme	32	uimsbf
asse_ID_length	8	Uimsbf
for(i=0;i<asset_ID_length;i++){		
asset_ID_byte	8	uimsbf
}		
number_of_items	8	uimsbf
for(i=0;i<number_of_items;i++){		
item_ID	32	uimsbf
item_tag	16	uimsbf
item_size	32	uimsbf
item_version	8	uimsbf
item_checksum	32	uimsbf
item_info_length	8	uimsbf
for(j=0;j<item_info_length;j++){		
item_info(i)		
}		
}		
descriptor_loop_length	16	uimsbf
for(i=0;i<descriptor_loop_length;i++){		
descriptor(i)		
}		
}		

【 図 2 6 】

データ・ロケーシヨん・マネヅメント・テーブル(DLMT)2600

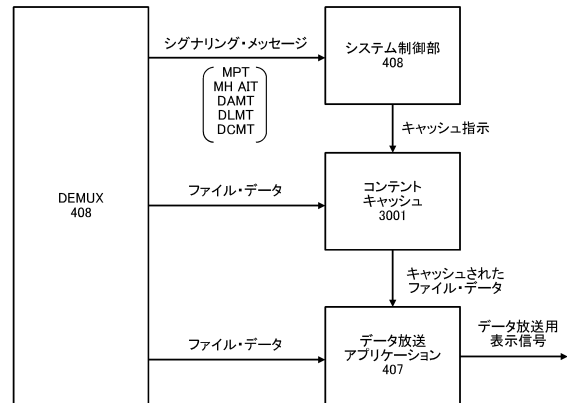
Syntax	No. of bit	Mnemonic
Data_Location_Management_Table(){		
table_id	8	uimsbf
version	8	uimsbf
length	16	uimsbf
base_URL_length	8	uimsbf
for(i=0;i<base_URL_length;i++){		
base_URL_byte	8	uimsbf
}		
number_of_items	8	uimsbf
for(i=0;i<number_of_items){		
item_tag	16	uimsbf
number_of_item_URI_bytes	8	uimsbf
for(l=0;l<number_of_item_URI_bytes;l++){		
item_URI_byte	8	uimsbf
}		
}		
}		

【 図 27 】

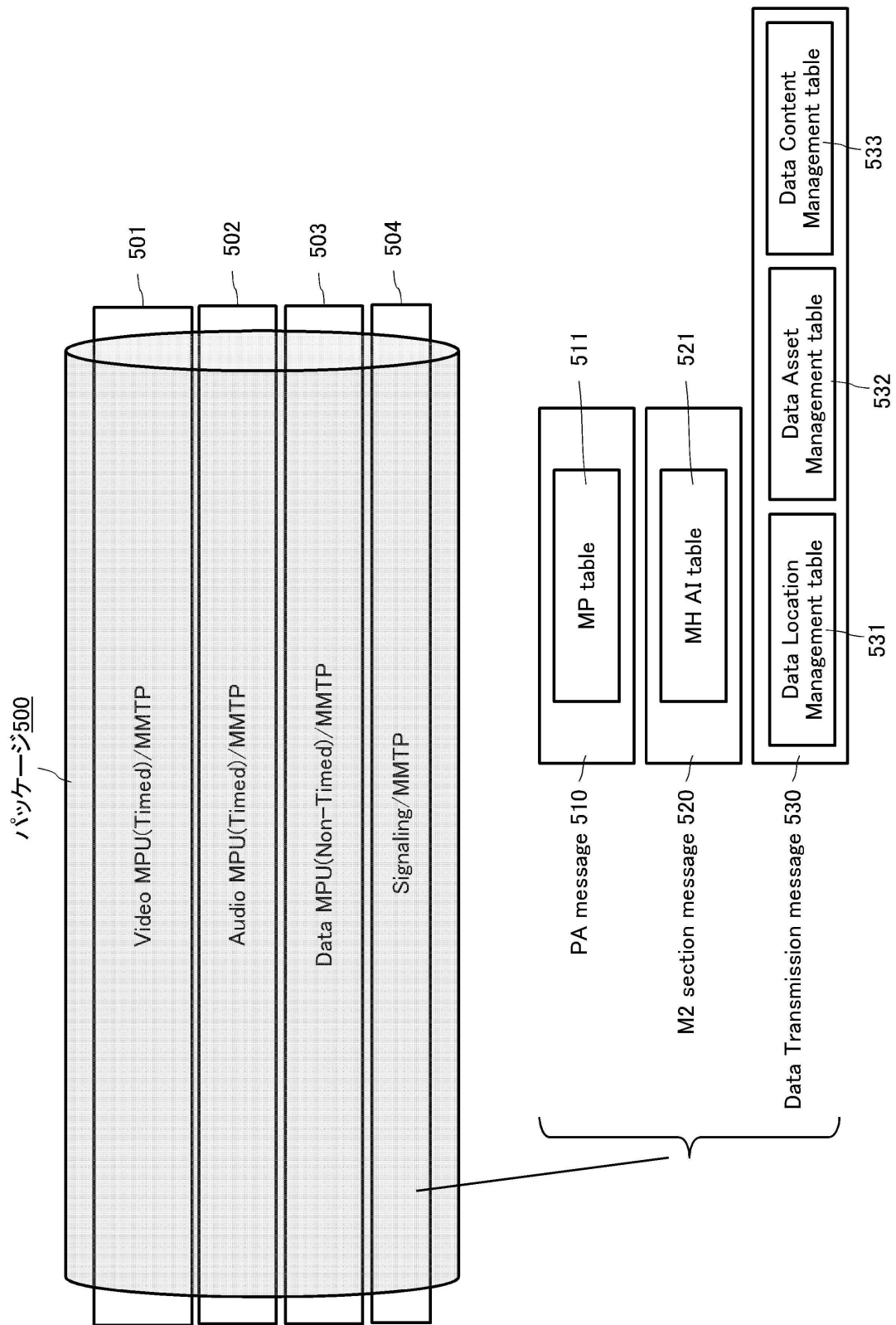
データ・コンテンツ・マネジメント・テーブル(DCMT)2700

Syntax	No. of bit	Mnemonic
Data_Content_Management_Table(){		
table_id	8	uimsbf
version	8	uimsbf
length	16	uimsbf
number_of_content	8	uimsbf
for(j=0;j<number_of_items;j++){		
contentID	16	uimsbf
content_version	8	uimsbf
content_cache_size	32	uimsbf
number_of_PU	8	Uimsbf
for(j=0;j<number_of_PU;j++){		
PU_tag	8	uimsbf
PU_cache_size	32	uimsbf
number_of_items	16	uimsbf
for(k=0;k<number_of_items;j++){		
item_tag	16	uimsbf
}		
number_of_linked_PU	8	uimsbf
for(k=0;k<number_of_items;k++){		
linked_PU_tag	8	uimsbf
}		
}		
}		

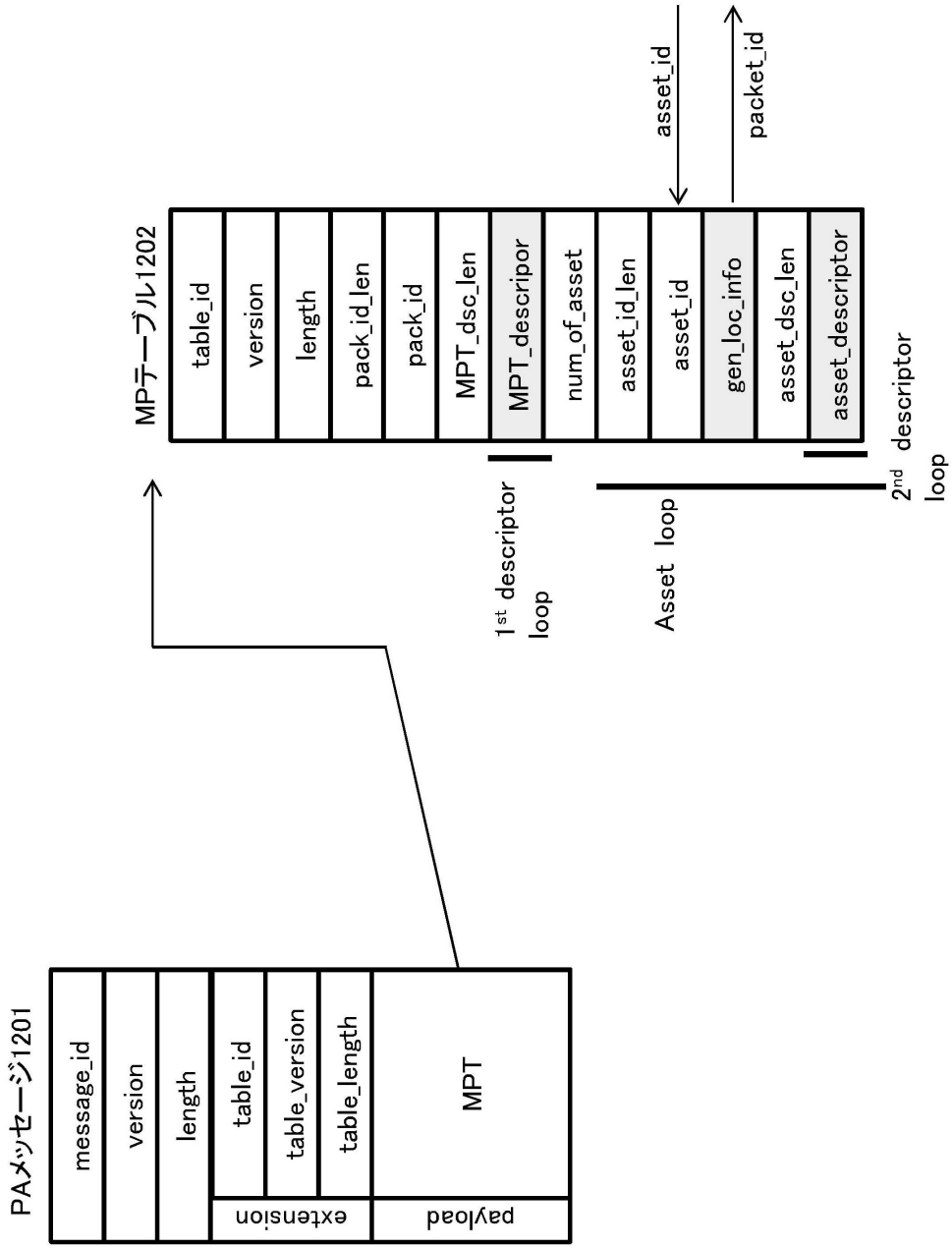
【 図 30 】



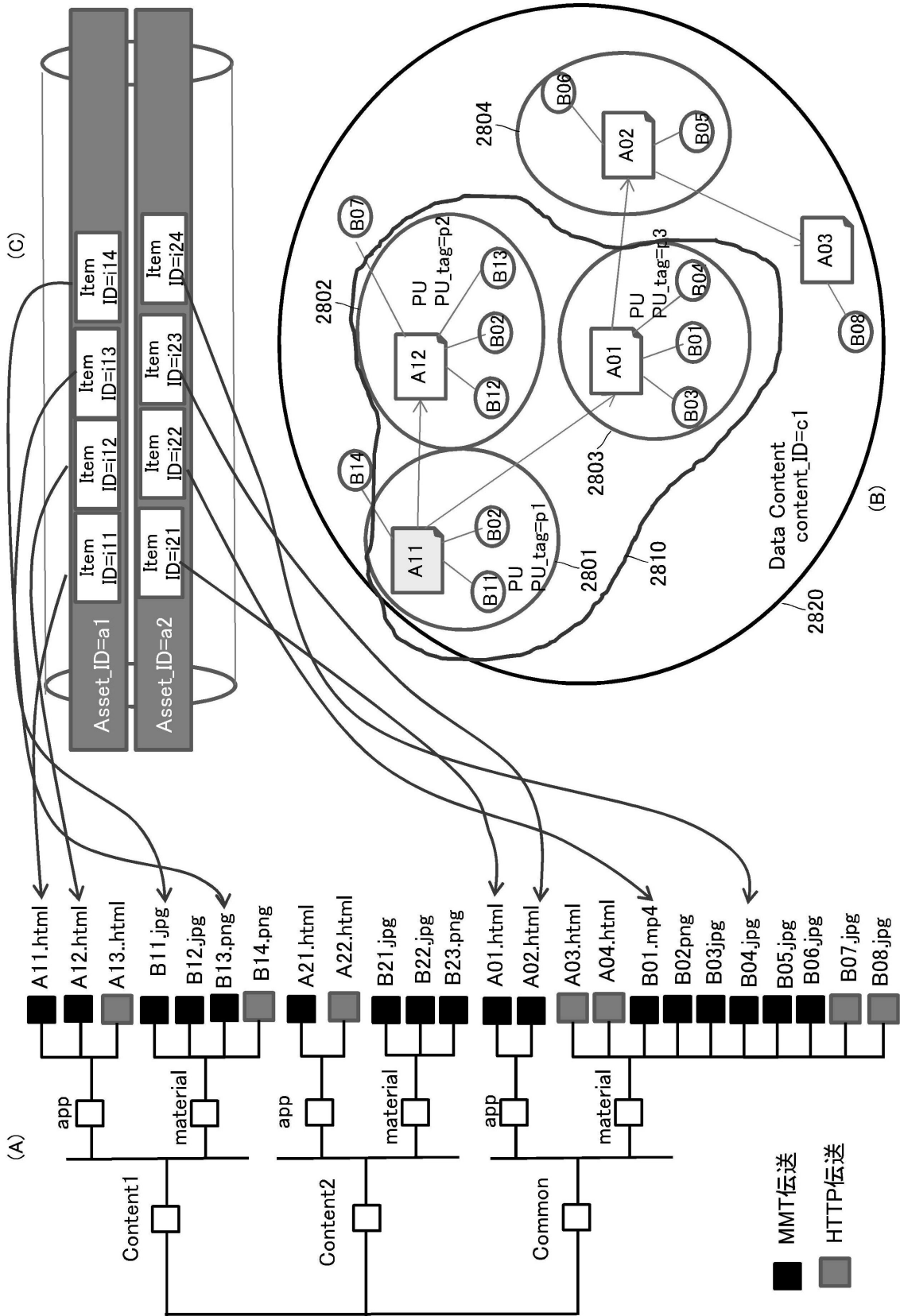
【 図 5 】



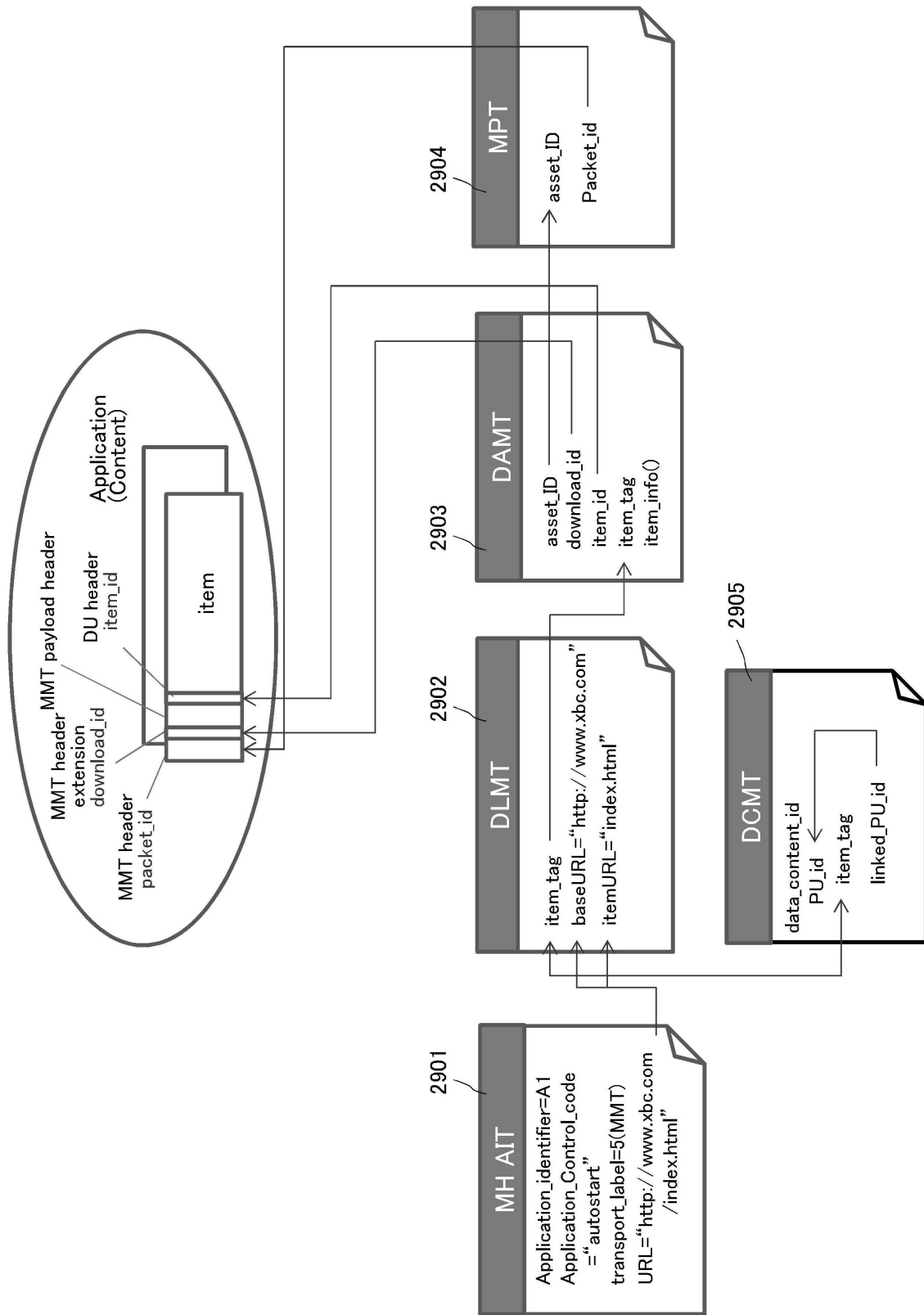
【 図 1 2 】



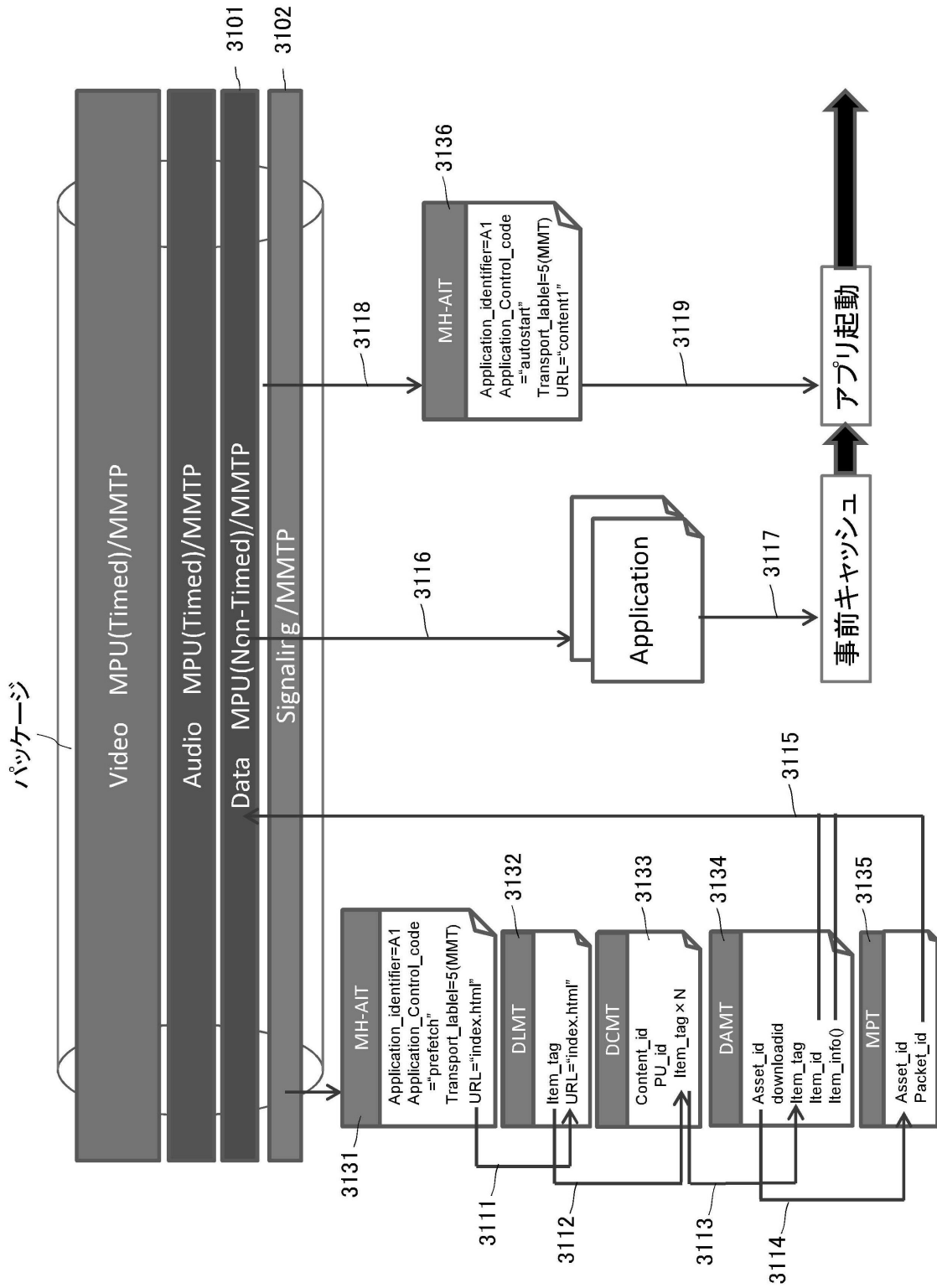
【 図 28 】



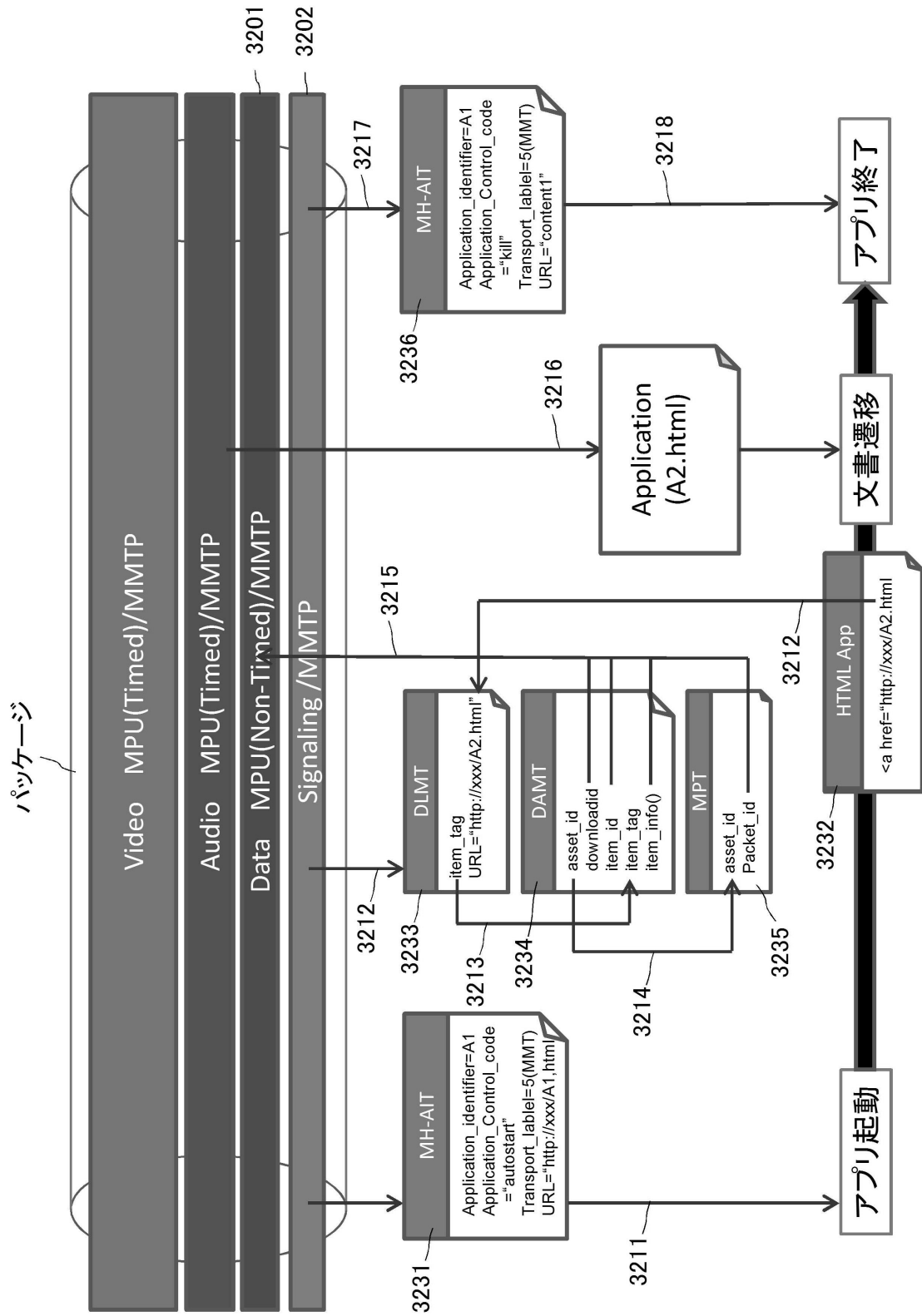
【 図 29 】

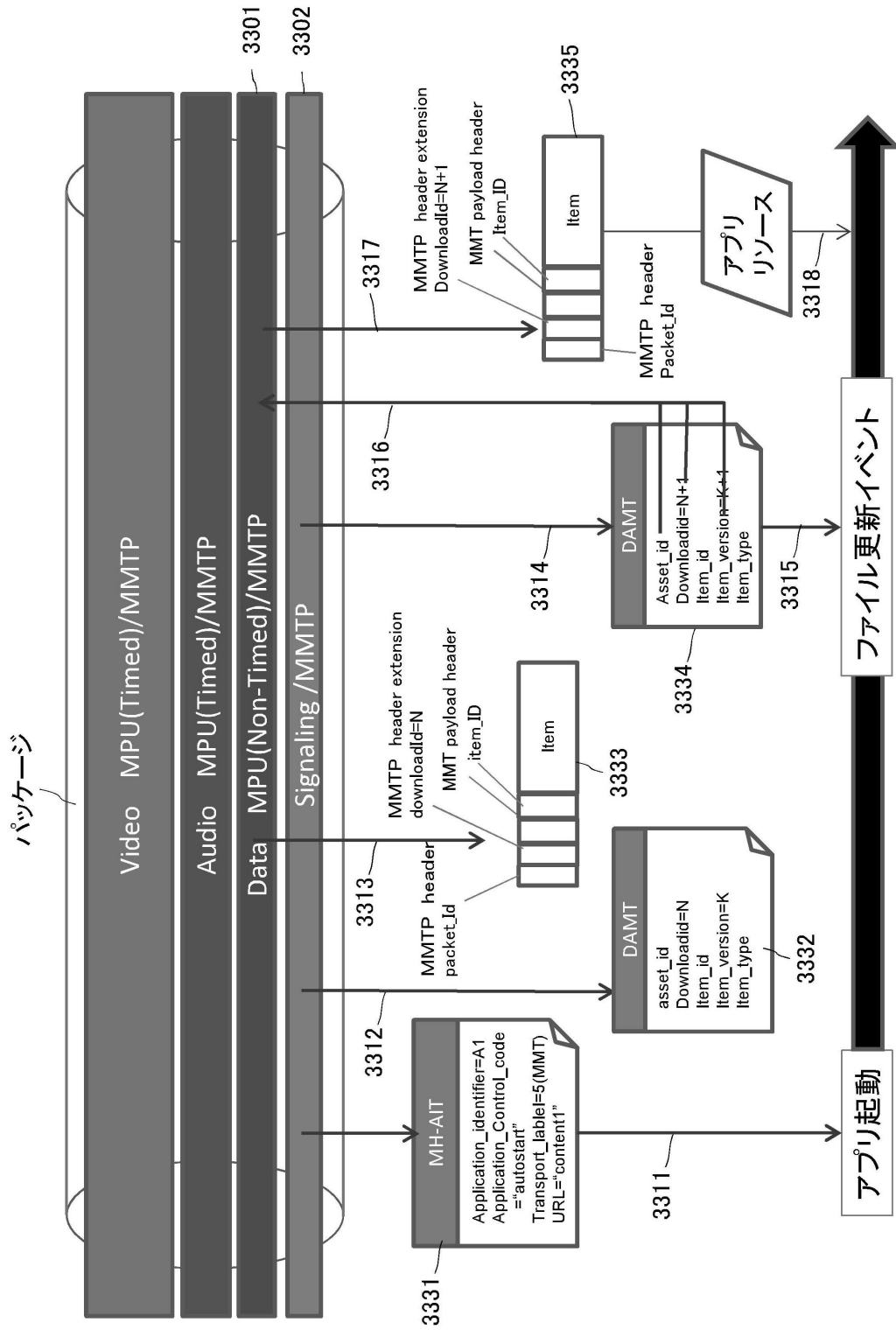


【 図 3 1 】



【 図 3 2 】





フロントページの続き

- (72)発明者 北里 直久
東京都港区港南1丁目7番1号 ソニー株式会社内
- (72)発明者 出葉 義治
東京都港区港南1丁目7番1号 ソニー株式会社内

審査官 後藤 嘉宏

- (56)参考文献 特開2001-186434(JP,A)
特開2000-36946(JP,A)
特開2000-31921(JP,A)
大槻ほか, スーパーハイビジョン衛星放送システムにおけるMMTを用いたデータ伝送方式の検討, 映像情報メディア学会技術報告, 日本, (一社)映像情報メディア学会, 2014年 2月 28日, Vol.38 No.14, p.29-32
大槻ほか, 放送・通信ハイブリッド配信における制御情報の提案, 電子情報通信学会2013年総合大会講演論文集 通信2, (一社)電子情報通信学会, 2013年 3月 5日, p.129
- (58)調査した分野(Int.Cl., DB名)
H04N 21/00 - 21/858