(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7248155号
(P7248155)

(45)発行日　令和5年3月29日(2023.3.29)

(24)登録日　令和5年3月20日(2023.3.20)

(51)国際特許分類

		FI	
H04N	*21/2362(2011.01)*	H04N	21/2362
H04N	*21/6543(2011.01)*	H04N	21/6543
H04N	*21/433(2011.01)*	H04N	21/433
H04N	*21/434(2011.01)*	H04N	21/434
H04H	*20/28 (2008.01)*	H04H	20/28

請求項の数　2　（全51頁）　最終頁に続く

(21)出願番号　特願2022-3176(P2022-3176)
(22)出願日　令和4年1月12日(2022.1.12)
(62)分割の表示　特願2020-217641(P2020-217641)の分割
原出願日　平成26年6月4日(2014.6.4)
(65)公開番号　特開2022-44657(P2022-44657A)
(43)公開日　令和4年3月17日(2022.3.17)
審査請求日　令和4年2月10日(2022.2.10)

(73)特許権者　000002185
ソニーグループ株式会社
東京都港区港南１丁目７番１号
(74)代理人　100093241
弁理士　宮田 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
弁理士法人大同特許事務所
(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
最終頁に続く

(54)【発明の名称】　受信装置

(57)【特許請求の範囲】
【請求項１】
アプリケーションを構成するファイルと、前記アプリケーションの伝送に関わるシグナリング・メッセージを取得する制御部と、
前記アプリケーションを構成するファイルがキャッシュされるメモリー部と、
を具備し、
前記制御部は、キャッシュへのロック対象ファイルを指定する情報をさらに取得し、前記メモリー部の容量と、前記キャッシュへのロック対象ファイルを指定する情報と、前記シグナリング・メッセージに含まれる、前記アプリケーションを構成するファイルのディレクトリーのロケーション情報と前記ディレクトリーを識別するノード・タグを記載する第１のテーブルと、前記ノード・タグを記載する第２のテーブルとに基づいて、前記アプリケーションを構成するファイルのキャッシュを制御する、
受信装置。
【請求項２】
前記制御部は、キャッシュされた前記アプリケーションを構成するファイルを前記メモリー部から削除する、
請求項１に記載の受信装置。
【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、所定のトランスポート方式によりデータ放送用のファイルを送信する送信装置及び送信方法、並びに、所定のトランスポート方式により伝送されるデータ放送用のファイルを受信する受信装置及び受信方法に関する。

【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）。次世代のディ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。ここで言うタイムド・メディアは、ビデオやオーディオ、字幕などの放送番組本編のストリーム・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）文書のようなデータ放送アプリケーション（コンテント）のファイル・データである。

【０００４】

放送番組に連動したデータ放送は、タイムリーな提示を行なうことが要求される。一方、データ放送で利用される各ファイルは、限られた放送伝送帯域で繰り返し伝送される。受信端末は、データ放送用のファイルをキャッシュしておくことによりデータ放送のタイムリーな提示を実現することができる。しかしながら、潤沢なキャッシュ・メモリーを装備しない受信端末においては、必要なファイルをキャッシュ・ミスすると、次の繰返し周期まで待たなければならなくなり、データ放送を提示するまで例えば数十秒程度の遅延を生じてしまう。

【０００５】

本出願時に運用されているＢＭＬ（Ｂｒｏａｄｃａｓｔ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ）によるデータ放送サービスでは、スクリプトから「ＬｏｃｋＭｏｄｕｌｅＯｎＭｅｍｏｒｙ（）」というＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ Ｐｒｏｇｒａｍｍｉｎｇ Ｉｎｔｅｒｆａｃｅ）を呼び出すことにより、特定のファイルをあらかじめキャッシュ・メモリーにプリキャッシュして留めておくことが可能である（例えば、特許文献２を参照のこと）。

【先行技術文献】

【特許文献】

【０００６】

【文献】特開２０１３－１５３２９１号公報
特開２００７－２７４１９３号公報

【非特許文献】

【０００７】

【文献】ＩＳＯ／ＩＥＣ ＦＤＩＳ ２３００８－１：２０１３（Ｅ） Ｉｎｆｏｒｍａｔｉｏｎ ｔｅｃｈｎｏｐｌｏｇｙ－Ｈｉｇｈ ｅｆｆｉｃｉｅｎｃｙ ｃｏｄｉｎｇ ａｎｄ ｍｅｄｉａ ｄｅｌｉｖｅｒｙ ｉｎ ｈｅｔｅｒｏｇｅｎｅｏｕｓ ｅｎｖｉｒｏｎｍｅｎｔｓ－Ｐａｒｔ１：ＭＰＥＧ ｍｅｄｉａ ｔｒａｎｓｐｏｒｔ（ＭＭＴ）

【発明の概要】

【発明が解決しようとする課題】

【０００８】

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によりデータ放送用のフ

ァイルを好適に送信することができる送信装置及び送信方法を提供することにある。
【０００９】
本明細書で開示する技術のさらなる目的は、所定のトランスポート方式により伝送されるデータ放送用のファイルを好適に受信することができる受信装置及び受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】
【００１０】
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、
データ放送で利用されるファイル・データを送信するファイル・データ送信部と、
データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ送信部と、
を具備する送信装置である。
【００１１】
本願の請求項２に記載の技術によれば、請求項１に記載の送信装置の前記シグナリング・メッセージ送信部は、データ放送提示単位毎に、提示単位を構成する放送伝送ファイル・リストと中心となるファイル、及び、プリキャッシュの対象ファイル・リストの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを送信するように構成されている。
【００１２】
本願の請求項３に記載の技術によれば、請求項１に記載の送信装置の前記シグナリング・メッセージ送信部は、データ放送提示単位毎に、提示単位を構成する放送伝送ファイル・リストと中心となるファイル、キャッシュにロックする対象ファイル及びロック対象のうちアンロックする対象ファイルの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを送信するように構成されている。
【００１３】
本願の請求項４に記載の技術によれば、請求項１乃至３のいずれかに記載の送信装置は、データ放送が連動する放送番組本体のメディア・データを送信するメディア・データ送信部をさらに備えている。
【００１４】
また、本願の請求項５に記載の技術は、
データ放送で利用されるファイル・データを送信するファイル・データ送信ステップと、
データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ送信ステップと、
を有する送信方法である。
【００１５】
また、本願の請求項６に記載の技術は、
データ放送で利用されるファイル・データを送信するファイル・データ受信部と、
データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ受信部と、
前記強制キャッシュ情報に基づいて前記ファイル・データ受信部が受信するファイル・データのキャッシュ・メモリーへ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。
【００１６】
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シグナリング・メッセージ受信部は、データ放送提示単位毎に、提示単位を構成する放送伝送ファイル・リストと中心となるファイル、及び、プリキャッシュの対象ファイル・リストの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを受信するように構成されている。
【００１７】

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制御部は、前記プリキャッシュの対象ファイル・リストに含まれているファイルを前記ファイル・データが受信すると、前記キャッシュ・メモリーにプリキャッシュするように構成されている。

【００１８】

本願の請求項９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シグナリング・メッセージ受信部は、データ放送提示単位毎に、提示単位を構成する放送伝送ファイル・リストと中心となるファイル、キャッシュにロックする対象ファイル及びロック対象のうちアンロックする対象ファイルの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを受信するように構成されている。

【００１９】

本願の請求項１０に記載の技術によれば、請求項９に記載の受信装置の前記制御部は、前記ロック対象のファイルを前記ファイル・データが受信すると、前記キャッシュ・メモリーにプリキャッシュするように構成されている。

【００２０】

本願の請求項１１に記載の技術によれば、請求項９に記載の受信装置の前記制御部は、前記アンロック対象のファイルを前記キャッシュ・メモリーから削除するように構成されている。

【００２１】

本願の請求項１２に記載の技術によれば、請求項６又は９のいずれかに記載の受信装置の前記制御部は、現在のデータ放送提示単位を構成する放送伝送ファイル・リストと中心となるファイルを前記ファイル・データが受信すると、前記キャッシュ・メモリーにキャッシュするように構成されている。

【００２２】

本願の請求項１３に記載の技術によれば、請求項６乃至１２のいずれかに記載の受信装置は、ファイル・データを利用してデータ放送を提示するデータ放送提示部をさらに備えている。

【００２３】

本願の請求項１４に記載の技術によれば、請求項６乃至１３のいずれかに記載の受信装置は、データ放送が連動する放送番組本体のメディア・データを受信するメディア・データ受信部と、メディア・データに基づいて放送番組を提示する放送番組提示部をさらに備えている。

【００２４】

また、本願の請求項１５に記載の技術は、

データ放送で利用されるファイル・データを送信するファイル・データ受信ステップと、

データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ受信ステップと、

前記強制キャッシュ情報に基づいて前記ファイル・データ受信部が受信するファイル・データのキャッシュ・メモリーへ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。

【発明の効果】

【００２５】

本明細書で開示する技術は、例えばＨＴＭＬ（Ｈｙｐｅｒ　Ｔｅｘｔ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ）５によるデータ放送のファイルをＭＭＴ方式で伝送することを想定している。本明細書で開示する技術によれば、放送局などの送信装置は、データ放送に関わるシグナリングに、強制キャッシュを指定する情報を含めて伝送することができる。また、本明細書で開示する技術によれば、家庭内に設置されたＳＴＢ（ＳｅｔＴｏｐＢｏｘ）やテレビ受信端末などの受信装置は、受信したデータ放送に関わるシグナリングに含まれている強制キャッシュ情報に基づいてデータ放送用の各ファイルのキャッシュ制御を好適に行ない、放送番組に連動したデータ放送をタイムリーに提示することができる。

【００２６】
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。
【００２７】
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】
【００２８】
【図１】図１は、本明細書で開示する技術を適用したディ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。
【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。
【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。
【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ー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レクター・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。
【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・ディレクトリー・マネジメント・テーブル（ＤＤＭＴ）２６００の構成例を示した図である。
【図２７】図２７は、データ・コンテント・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、データ・コンテント・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２９】図２９は、ＭＭＴ伝送されるデータ放送アプリケーション（コンテント）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３０】図３０は、ＭＭＴ伝送路からデータ放送アプリケーション（コンテント）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。
【図３１】図３１は、受信機内でデータ放送アプリケーション（コンテント）をキャッシュする仕組みを模式的に示した図である。
【図３２】図３２は、受信機１２における方式１に基づくファイル・データのキャッシュ制御手順を示したフローチャートである。
【図３３】図３３は、受信機１２における方式１に基づくファイル・データのプリキャッシュ動作例を示した図である。
【図３４】図３４は、受信機１２における方式２に基づくファイル・データのキャッシュ制御手順を示したフローチャートである。
【図３５】図３５は、受信機１２における方式２に基づくファイル・データのキャッシュのロック及びアンロック動作例を示した図である。
【発明を実施するための形態】
【００２９】
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。
【００３０】
図１には、本明細書で開示する技術を適用したディジタル放送システム１０の構成例を模式的に示している。図示のディ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。
【００３１】
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、字幕などの放送番組本編に関わるストリーム・データである。また、ノンタイムド・メディアは、例えばＨＴＭＬ文書のような、データ放送に利用される各ファイル・データである。以下の説明では、ＨＴＭＬ５によるデータ放送サービスを想定している。
【００３２】
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。また、受信機１２は、受信した放送信号からデータ放送用の各ファイル・データを取得すると、ＨＴＭＬブラウザーなどのアプリケーション・エンジンを起動して、放送番組に連動したデータ放送の提示を行なう。
【００３３】
図１に示したディ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。
【００３４】
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。
【００３５】

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。

【００３６】

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。

【００３７】

ＭＭＴパケット２０７のＭＭＴペイロード２０９には、ＭＦＵ（ＭＭＴ　Ｆｒａｇｍｅｎｔ　Ｕｎｉｔ）２１０あるいはデータ放送に関わるシグナリング・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ーである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。

【００３８】

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。

【００３９】

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。

【００４０】

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・メディアであるデータ放送用の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。

【００４１】

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・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。

【００４２】

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データ放送に関わるシグナリング・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、データ放送に関わるシグナリング・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・メッセージの３種類に大別される。本実施形態では、データ放送で利用される各ファイルの強制キャッシュを指定する情報がデータ・トランスミッション・メッセージ内に含まれる。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。

【００４３】

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ト（データ放送用アプリケーション）を構成するものである。

【００４４】

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。

【００４５】

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。

【００４６】

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。

【００４７】

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。

【００４８】

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。

【００４９】

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。

【００５０】

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。

【００５１】

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パケットがＩＰサービス・マルチプレクサー３１０に送られる。

【００５２】

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データ放送に関わるシグナリング・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。
【００５３】
　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。
【００５４】
　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。
【００５５】
　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。
【００５６】
　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変調処理が行なわれ、そのＲＦ変調信号がＲＦ伝送路に送出される。
【００５７】
　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、アプリケーション・データ制御部４０７と、キャッシュ・メモリー４０８と、データ放送アプリケーション・エンジン４０９と、システム制御部４１０と、合成部４１１と、ＩＰインターフェース４１２を備えている。
【００５８】
　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で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、放送番組本編に関わるビデオ、オーディオ、キャプションの各符号化信号、放送番組に連動するデータ放送に利用されるファイル・データ、並びにシグナリング情報を出力する。なお、データ放送に利用されるファイル・データは、例えばＨＴＭＬ５形式で記述されたデータ放送アプリケーションである。
【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】
　アプリケーション・データ制御部４０７は、データ放送で利用される各ファイル・データの処理を行なう。本実施形態では、データ放送で利用される各ファイル・データは放送信号並びにＩＰネットワークの２系統から伝送されることを想定し、前者はチューナー・復調部４０１及びデマルチプレクサー４０２経由で、後者はＩＰインターフェース４１２経由で、それぞれアプリケーション・データ制御部４０７が取得する。アプリケーション・データ制御部４０７は、デマルチプレクサー４０２から出力されるシグナリング情報に

基づいて、取得したファイル・データの処理を制御する。具体的には、アプリケーション・データ制御部４０７は、現在いずれのデータ放送提示単位（Ｐｒｅｓｅｎｔａｔｉｏｎ Ｕｎｉｔ：ＰＵ）にいるかを管理し、該当するファイル・データ（ＨＴＭＬ５などのデータ放送アプリケーション）の処理をＨＴＭＬブラウザーなどのデータ放送アプリケーション・エンジン４０９に指示する。データ放送アプリケーション・エンジン４０９は、適宜キャッシュ・メモリー４０８に事前キャッシュ又はプリキャッシュされたファイル・データを用いて、データ放送アプリケーションを処理する。

【００６１】

アプリケーション・データ制御部４０７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・データを事前にキャッシュ・メモリー４０８にキャッシュ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。

【００６２】

また、データ放送で利用される各ファイルの強制キャッシュを指定する情報がデータ・トランスミッション・メッセージ内に含まれている。本実施形態では、アプリケーション・データ制御部４０７は、データ・トランスミッション・メッセージ内に含まれている強制キャッシュ情報に基づいて、データ放送をタイムリーに提示するために必要なファイル・データをキャッシュ・メモリー４０８にプリキャッシュする。ファイル・データのプリキャッシュの詳細については後述に譲る。

【００６３】

なお、放送ストリームでは、同一コンテントのファイル・データが繰り返し送られてくる。システム制御部４１０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アプリケーション・データ制御部４０７で取得されるようにする。

【００６４】

システム制御部４１０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。

【００６５】

また、システム制御部４１０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４１１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。ビデオ信号及びオーディオ信号からなる放送番組本編は、図示しないモニター・ディスプレイから映像及び音声出力される。また、データ放送アプリケーション・エンジン４０９が処理したデータ放送も、モニター・ディスプレイ上で放送番組本編の画面に重畳して表示される。

【００６６】

図４に示した受信機１２の動作について説明しておく。

【００６７】

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で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。

【００６８】
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。
【００６９】
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４１０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。
【００７０】
そして、合成部４１１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。
【００７１】
また、デマルチプレクサー４０２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４０５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信号が得られる。ビデオ信号及びオーディオ信号からなる放送番組本編は、図示しないモニター・ディスプレイから映像及び音声出力される。
【００７２】
一方、アプリケーション・データ制御部４０７は、現在いずれのデータ放送提示単位にいるかを管理し、該当するファイル・データ（ＨＴＭＬ５などのデータ放送アプリケーション）の処理をＨＴＭＬブラウザーなどのデータ放送アプリケーション・エンジン４０９に指示する。データ放送アプリケーション・エンジン４０９は、適宜キャッシュ・メモリー４０８に事前キャッシュ又はプリキャッシュされたファイル・データを用いて、データ放送アプリケーションを処理する。データ放送アプリケーション・エンジン４０９が処理したデータ放送も、モニター・ディスプレイ上で放送番組本編の画面に重畳して表示される。
【００７３】
また、アプリケーション・データ制御部４０７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、キャッシュ・メモリー４０８の空き容量に応じたファイル・データの事前キャッシュや、放送番組に連動してデータ放送をタイムリーに提示するための強制キャッシュを制御する（後述）。
【００７４】
図１に示したディ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５には、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５００のイメージを示している。
【００７５】
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、字幕などの放送番組本編に関わるタイムド・メディアと、放送番組に連動するデータ放送に利用されるファイル・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５００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５０１～５０４が利用されている。なお、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。

【００７６】

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（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（アセット識別子）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ーである）。すなわち、伝送路５０１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５０２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５０３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。

【００７７】

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・コンテント（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・コンテントには別々の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５０３を１本しか描いていない。

【００７８】

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、ＩＰネットワークなど通信の伝送路を介して提供することもできる。

【００７９】

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・ディレクトリー・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・コンテント・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。

【００８０】

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。

【００８１】
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。
【００８２】
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。
【００８３】
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。
【００８４】
【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

【００８５】
参照番号６０５で示すＲＡＰ（Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｐｏｉｎｔ）フラグに１が代入されていると、当該ＭＭＴＰ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Ｒａｎｄｏｍ　Ａｃｃｅｓｓ　Ｐｏｉｎｔを含んでいることを表す。
【００８６】
参照番号６０７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ＭＭＴＰ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ＭＭＴパッケージ（ＭＰ）テーブルで示されている。
【００８７】
参照番号６０８で示す、３２ビット長のｔｉｍｅｓｔａｍｐフィールドには、当該ＭＭＴＰパケットの送信時間が、ＮＴＰプロトコルで規定されているｓｈｏｒｔ－ｆｏｒｍａｔで記載される。
【００８８】
参照番号６０９で示す、３２ビット長のｐａｃｋｅｔ＿ｓｅｑｕｅｎｃｅ＿ｎｕｍｂｅ

ｒ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（ＭＭＴ伝送路上でのシーケンス番号）が記載される。

【００８９】

図７には、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の構成例を示している。図示のように、この場合、ｌｅｎｇｔｈフィールド７０１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７０２には、ｄｏｗｎｌｏａｄ＿ｉｄが４バイトで記載される。

【００９０】

ＭＭＴプロトコルを使ってＭＰＵ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、ＭＰＵはＭＭＴＰペイロードに挿入され、ＭＭＴＰパケットで伝送される。ＭＭＴＰ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ように、ＭＭＴＰペイロードのフラグメンテーションを許容する。また、ＭＴＰ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ＭＭＴＰペイロードを単一のＭＭＴＰ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ＭＰＵデータを復元する。

【００９１】

図８には、ＭＰＵモードの場合のＭＭＴＰペイロード８００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。ＭＰＵモードは、ＭＭＴＰヘッダーのｔｙｐｅフィールド６０６に「０ｘ００」が書き込まれている場合である。ＭＰＵモードのＭＭＴＰパケットは、放送番組本編に関わるビデオ、オーディオ、並びに、放送番組に連動するファイル・データ（データ放送アプリケーション）の伝送に使用される。

【００９２】

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。

【００９３】

【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.

【００９４】

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。

【００９５】

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。

【００９６】

【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

【００９７】
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。
【００９８】
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。
【００９９】
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。
【０１００】
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＿Ｈｅａｄｅｒ８０７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。
【０１０１】
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＿Ｈｅａｄｅｒ９００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＿Ｈｅａｄｅｒ１０００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＿Ｈｅａｄｅｒ１０００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。
【０１０２】
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。
【０１０３】
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。
【０１０４】
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・サイズが大きくないので、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。

【０１０５】

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。

【０１０６】

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。

【０１０７】

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

【０１０８】

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。

【０１０９】

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。

【０１１０】

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。

【０１１１】

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。

【０１１２】

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ーの説明を示している。

【０１１３】

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ーである。例えばＭＰ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、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ーである。

【０１１４】

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。

【０１１５】

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。

【０１１６】

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ー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。

【０１１７】

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、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ーである。

【０１１８】

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・データ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。

【０１１９】

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。ＭＰＴ＿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ーである。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。
【０１２０】
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ーである。ｎｕｍｂｅｒ＿ｏｆ＿ａｓｓｅｔの数分（Ｎ３）だけ、以下のＡｓｓｅｔ　ｌｏｏｐが配置される。
【０１２１】
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の各パラメーターが配置される。
【０１２２】
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。
【０１２３】
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＿ｄｅｓｃｒｉｐｔｏｒ＿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。
【０１２４】
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。
【０１２５】
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＿ｅｘｔｅｎｔｉｏｎ（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。
【０１２６】
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。
【０１２７】
　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・シンタクス指示）は、１ビットのフィールドで、常に「１」とする。ｓｅｃｔｏｉｎ＿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の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。ａｐｐｌｉｃａｔｏｎ＿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。
【０１２８】
　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。
【０１２９】
　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。
【０１３０】
　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。
【０１３１】
　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ーである。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＿ｃｏｄ

ｅ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。

【０１３２】

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。

【０１３３】

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ー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の各パラメーターの意味について説明する。

【０１３４】

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。

【０１３５】

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り、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。

【０１３６】

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。

【０１３７】

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。

【０１３８】

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。

【０１３９】

ｐｒｏｔｏｃｏｌ＿ｉｄ（プロトコル識別子）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。

【０１４０】

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。

【０１４１】

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。

【０１４２】

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、ＵＲＬ＿ｂａｓｅが"ｈｔｔｐ：／／ｗｗｗ．ｘｂｃ．ｃｏｍ"で、ＵＲＬ＿ｅｘｔｅｎｓｉｏｎが"ｉｎｄｅｘ．ｈｔｍｌ"であれば、これらの文字列を連結して、完全なＵＲＬ"ｈｔｔｐ：／／ｘｂｃ．ｃｏｍ／ｉｎｄｅｘ．ｈｔｍｌ"を得ることができる。

【０１４３】

要するに、ＭＨ　ＡＩ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。

【０１４４】

図２４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２４００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。

【０１４５】

ｍｅｓｓａｇｅ＿ｉｄ（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１６ビットの固定値であり、本実施形態では０ｘＦ０００とする。ｖｅｒｓｉｏｎ（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。ｌｅｎｇｔｈ（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、３２ビットのパラメーターである。

【０１４６】

ｎｕｍ＿ｏｆ＿ｔａｂｌｅｓ（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、ｎｕｍ＿ｏｆ＿ｔａｂｌｅｓが示す数分だけ、テーブル情報のループが配置される。

【０１４７】

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、ｔａｂｌｅ＿ｉｄ（テーブル識別）、ｔａｂｌｅ＿ｖｅｒｓｉｏｎ（テーブル・バージョン）、並びに、ｔａｂｌｅ＿ｌｅｎｇｔｈ（テーブル長）が格納される。ｔａｂｌｅ＿ｉｄ（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・メッセージでは、データ・アセット・マネジメント・テーブル（ＤＡＭＴ）、データ・ディレクトリー・マネジメント・テーブル（ＤＤＭＴ）、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）の３種類のシグナリング・テーブルが伝送されるが（前述）、ｔａｂｌｅ＿ｉｄ（テーブル識別子）はこれらのうちいずれのテーブルであるかを識別する。ｔａｂｌｅ＿ｖｅｒｓｉｏｎ（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。ｔａｂｌｅ＿ｌｅｎｇｔｈ（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。

【０１４８】

また、ｎｕｍ＿ｏｆ＿ｔａｂｌｅｓ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、ｔａｂｌｅ＿ｉｄで識別されるテーブルの中身の情報が格納される。ｔａｂｌｅ（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。

【０１４９】

図２５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２５００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。

【０１５０】

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ーである。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・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ーである。

【０１５１】

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ーである。ｎｕｍｂｅｒ＿ｏｆ＿ａｓｓｅｔの数分だけ、以下のアセット情報のループが配置され、アセット毎のファイル・データの情報が格納される。

【０１５２】

１つのアセット情報のループ内には、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ＭＭＴＰ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）。

【０１５３】

アセット情報のループ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、ａｓｓｅｔ＿ＩＤの形式を示す。ａｓｓｅｔ＿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、ａｓｓｅｔ＿ＩＤ＿ｂｙｔｅ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

、ａｓｓｅｔ＿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。

【０１５４】

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。

【０１５５】

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、ｎｏｄｅ＿ｔａｇ、ｉｔｅｍ＿ｓｉｚｅ、ｉｔｅｍ＿ｖｅｒｓｉｏｎ、ｉｔｅｍ＿ｃｈｅｃｋｓｕｍ、ｉｔｅｍ＿ｉｎｆｏの各パラメーターが記述される。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。ｎｏｄｅ＿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ｎｏｄｅ＿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。なお、ｎｏｄｅは、データ放送アプリケーション（コンテント）を構成するディレクトリー構造上のノードとなるディレクトリー並びにアイテムの各々を指す。アイテムだけでなくディレクトリーもｎｏｄｅ＿ｔａｇで指定することができる。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＋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ｃｈｅｃｋ＿ｓｕｍ＿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＿ｃｈｅｃｋ＿ｓｕｍ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（ｉｔｅｍ＿ｉｎｆｏ（））を格納する。

【０１５６】

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の情報（ｄｅｓｃｒｉｐｔｏｒ（））を格納する。格納される記述子は別途定義する。

【０１５７】

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（コンテント）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、ｎｏｄｅ＿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ＭＭＴ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ｎｏｄｅ＿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。

【０１５８】

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・ディレク

トリー・マネジメント・テーブル（ＤＤＭＴ）２６００の構成例を示している。データ・ディレクトリー・マネジメント・テーブルは、データ放送アプリケーション（コンテント）を構成するディレクトリー、並びに、ディレクトリーに含まれる各ノード（下位のディレクトリーやアイテム（ファイル・データ））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・ディレクトリー・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。

【０１５９】

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・ディレクトリー・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・ディレクトリー・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・ディレクトリー・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・ディレクトリー・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。

【０１６０】

ｂａｓｅ＿ｆｏｌｄｅｒ＿ｐａｔｈ＿ｌｅｎｇｔｈは、ｂａｓｅ＿ｆｏｌｄｅｒ＿ｐａｔｈ＿ｂｙｔｅの情報領域のサイズをバイト単位で表す。ｂａｓｅ＿ｆｏｌｄｅｒ＿ｐａｔｈ＿ｂｙｔｅは、ｂａｓｅ＿ｆｏｌｄｅｒ＿ｐａｔｈ＿ｌｅｎｇｔｈの数分のループからなる一連の領域に、ｂａｓｅ＿ｆｏｌｄｅｒ（上位のディレクトリー）へのパス名を格納する。ｂａｓｅ＿ｆｏｌｄｅｒ＿ｐａｔｈ＿ｂｙｔｅは、例えば、対応するディレクトリーへアクセスするための絶対的なＵＲＬ形式で表記される。

【０１６１】

ｎｕｍ＿ｏｆ＿ｆｏｌｄｅｒ＿ｎｏｄｅｓは、データ・ディレクトリー・マネジメント・テーブルに記載されるフォルダー・ノードの数を示す。そして、ｎｕｍ＿ｏｆ＿ｆｏｌｄｅｒ＿ｎｏｄｅｓの数分だけフォルダー・ノードのループが配置される。

【０１６２】

１つのフォルダー・ノードのループ内には、データ・ディレクトリー・マネジメント・テーブルに記載される各フォルダー・ノードの情報と、ベース・フォルダーに含まれる各ファイル・データの情報が格納される。

【０１６３】

ｆｏｌｄｅｒ＿ｎｏｄｅ＿ｐａｔｈ＿ｌｅｎｇｔｈは、ｆｏｌｄｅｒ＿ｎｏｄｅ＿ｐａｔｈ＿ｂｙｔｅの情報領域のサイズをバイト単位で表す。ｆｏｌｄｅｒ＿ｎｏｄｅ＿ｐａｔｈ＿ｂｙｔｅは、ｆｏｌｄｅｒ＿ｎｏｄｅ＿ｐａｔｈ＿ｌｅｎｇｔｈの数分のループからなる一連の領域に、ｆｏｌｄｅｒ＿ｎｏｄｅへのパス名を格納する。ｆｏｌｄｅｒ＿ｎｏｄｅ＿ｐａｔｈ＿ｂｙｔｅは、例えば、対応するディレクトリーへアクセスするための、ｂａｓｅ＿ｆｏｌｄｅｒ＿ｐａｔｈからの相対的なＵＲＬ形式で表記される。図示しないが、フォルダー・ノードの情報としてｆｏｌｄｅｒ＿ｎｏｄｅ＿ｖｅｒｓｉｏｎ（フォルダー・ノードのバージョン情報）を含んでいてもよい。例えば、ｂａｓｅ＿ｆｏｌｄｅｒのパス名（ＵＲＬ）が"ｈｔｔｐ：／／ｗｗｗ．ｘｂｃ．ｃｏｍ"で、あるｆｏｌｄｅｒ＿ｎｏｄｅのパス名（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：／／ｘｂｃ．ｃｏｍ／ｉｎｄｅｘ．ｈｔｍｌ"を得ることができる。

【０１６４】

ｎｕｍ＿ｏｆ＿ｆｉｌｅｓは、データ・ディレクトリー・マネジメント・テーブルに記載されるファイルの数を示す。そして、ｎｕｍ＿ｏｆ＿ｆｉｌｅｓの数分だけファイルのループが配置される。

【０１６５】

１つのファイルのループ内には、ベース・フォルダーに含まれる各ファイル・データの情報として、ｎｏｄｅ＿ｔａｇと、ｆｉｌｅ＿ｎａｍｅ＿ｂｙｔｅ（ファイル名）が格納

される。ｎｏｄｅ＿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よりも短い１６ビットで表す。なお、ｎｏｄｅは、データ放送アプリケーション（コンテント）を構成するディレクトリー構造上のノードとなるディレクトリー並びにアイテムの各々を指す。アイテムだけでなくディレクトリーもｎｏｄｅ＿ｔａｇで指定することができる（前述）。ｆｉｌｅ＿ｎａｍｅ＿ｂｙｔｅは、ｆｉｌｅ＿ｎａｍｅ＿ｌｅｎｇｔｈの数分のループからなる一連の領域に格納される。

【０１６６】

要するに、データ・ディレクトリー・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ディレクトリー並びにディレクトリーに含まれるサブディレクトリーやファイル（アイテム）に関するディレクトリー構造を管理するテーブルである。データ・ディレクトリー・マネジメント・テーブル２６００により、データ放送アプリケーションのファイル構成と、ファイル伝送のための構成を分離することができる。また、データ・ディレクトリー・マネジメント・テーブル２６００を参照して、ｎｏｄｅ＿ｔａｇから該当するアイテムのパス名（ＵＲＬ）を引いたり、逆にパス名（ＵＲＬ）から該当するｎｏｄｅ＿ｔａｇを引いたりすることができる。なお、本構成例では、データ・ディレクトリー・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ｆｏｌｄｅｒ＿ｐａｔｈ＿ｂｙｔｅとして設定するとともに各ディレクトリーをｎｏｄｅ＿ｔａｇとして識別情報を与え、アイテム毎の情報としてはファイル名とｎｏｄｅ＿ｔａｇのみを指定するので、ｆｏｌｄｅｒ＿ｐａｔｈ＿ｂｙｔｅの情報量が大きくなり過ぎるということはない。

【０１６７】

続いて、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・コンテント・マネジメント・テーブル（ＤＣＭＴ）について説明する。

【０１６８】

データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）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ト（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。本実施形態では、データ・コンテント・マネジメント・テーブルに強制キャッシュを指定する情報を含めて伝送する。放送番組に連動したデータ放送は、タイムリーな提示を行なうことが要求される。このような場合、放送送出システム１１側からは、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）に強制キャッシュを指定する情報を含めることで、受信機１２側では、放送番組に連動したデータ放送で利用される各ファイルをプリキャッシュしておき、データ放送のタイムリーな提示を実現することができる。

【０１６９】

データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）による強制キャッシュ情報の伝送方式として、以下の２通りを挙げることができる。

【０１７０】

（方式１）データ・コンテント・マネジメント・テーブルにおいて、データ放送提示単位（ＰＵ）毎に、提示単位を構成する放送伝送ファイル（ｍｅｍｂｅｒ　ｉｔｅｍ）リストと中心となるファイル（ｐｒｉｍａｒｙ　ｉｔｅｍ）、及びプリキャッシュの対象ファイルが存在する場合はその対象ファイル・リスト（ｐｒｅ－ｃａｃｈｅ　ｉｔｅｍ）の情報を記述する。

【０１７１】

ここで言うプリキャッシュの対象ファイルは、例えば、現在のデータ放送提示単位（ＰＵ）から次に参照するデータ放送提示単位（ＰＵ）を構成する放送ファイル・リストで構成される。放送番組の制作側では、このようにデータ放送のシグナリング情報でプリキャッシュの対象ファイル・リストを提示することで、受信機側では次に遷移するデータ放送提示単位に必要なファイル・データをプリキャッシュしておくという、プリキャッシュによるキャッシュ制御動作を行なうことができる。その結果、放送番組に連動したタイムリーなデータ放送サービスを実現することができる。

【０１７２】
（方式２）データ・コンテント・マネジメント・テーブルにおいて、データ放送提示単位（ＰＵ）毎に、提示単位を構成する放送伝送ファイル（ｍｅｍｂｅｒ　ｉｔｅｍ）リストと中心となるファイル（ｐｒｉｍａｒｙ　ｉｔｅｍ）、及びキャッシュにロックする対象ファイル（ｌｏｃｋ　ｃａｃｈｅ　ｉｔｅｍ）及びロック対象のうちアンロックする対象ファイル（ｕｎｌｏｃｋ　ｃａｃｈｅ　ｉｔｅｍ）の情報を記述する。
【０１７３】
ここで言うロック対象ファイルは、例えば、現在のデータ放送提示単位（ＰＵ）から次に参照するデータ放送提示単位（ＰＵ）で使用する放送ファイル・リストで構成される。放送番組の制作側では、このようにデータ放送のシグナリング情報でロック並びにアンロックの対象ファイル・リストを提示することで、受信機側ではロック並びにアンロックによるキャッシュ制御動作を行なうことができる。例えば、受信機側では、次に遷移するデータ放送提示単位に必要なファイル・データをキャッシュにロックしておくことができる。その結果、放送番組に連動したタイムリーなデータ放送サービスを実現することができる。また、アンロック対象ファイルは、例えば、現在のデータ放送提示単位（ＰＵ）では不要となる放送ファイル・リストで構成される。アンロック対象ファイルを指定することで、不要なファイルをキャッシュ・メモリー４０８から削除することができ、メモリー・サイズを節約することができる。
【０１７４】
図２７には、方式１を実現するデータ・トランスミッション・メッセージで伝送されるデータ・コンテント・マネジメント・テーブル（ＤＣＭＴ）２７００の構成例を示している。
【０１７５】
ｔａｂｌｅ＿ｉｄ（テーブル識別）には、各種シグナリング情報においデータ・コンテント・マネジメント・テーブルであることを示す８ビットの固定値が書き込まれる。ｖｅｒｓｉｏｎ＿（バージョン）は、このデータ・コンテント・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ーである。例えばデータ・コンテント・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ーである。
【０１７６】
ｎｕｍｂｅｒ＿ｏｆ＿ｃｏｎｔｅｎｔは、パッケージに含まれるコンテントの数を示す、８ビットのパラメーターである（コンテントは、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、以下のコンテントのループが配置され、コンテント毎の情報が格納される。
【０１７７】
１つのコンテントのループ内には、コンテントに関する情報として、ｃｏｎｔｅｎｔ＿ＩＤと、ｃｏｎｔｅｎｔ＿ｖｅｒｓｉｏｎと、ｃｏｎｔｅｎｔ＿ｃａｃｈｅ＿ｓｉｚｅと、当該コンテントに含まれるデータ放送提示単位（Ｐｒｅｓｅｎｔａｔｉｏｎ　Ｕｎｉｔ：ＰＵ）に関する情報が書き込まれる。ｃｏｎｔｅｎｔ＿ＩＤは、コンテントの識別情報である。ｃｏｎｔｅｎｔ＿ｖｅｒｓｉｏｎは、コンテントのバージョンを示す。ｃｏｎｔｅｎｔ＿ｃａｃｈｅ＿ｓｉｚｅは、コンテントをキャッシュするサイズを示す。
【０１７８】
ｎｕｍｂｅｒ＿ｏｆ＿ＰＵは、コンテントに含まれるデータ放送提示単位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。
【０１７９】
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と、当該データ放送提示単位（ＰＵ）の中

心となるファイル（ｐｒｉｍａｒｙ　ｉｔｅｍ）を識別するＰＵ＿ｐｒｉｍａｒｙ＿ｉｔｅｍ＿ｎｏｄｅ＿ｔａｇが書き込まれる。

【０１８０】

また、ＰＵのループ内には、当該データ放送提示単位（ＰＵ）を構成する放送伝送ファイル（ｍｅｍｂｅｒ　ｉｔｅｍ）リストが書き込まれる。具体的には、ｎｕｍｂｅｒ＿ｏｆ＿ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅｓは、当該データ放送提示単位（ＰＵ）に含まれる（すなわち、ＰＵのメンバーとなる）ノードの数を示す。その後に、このｎｕｍｂｅｒ＿ｏｆ＿ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅｓの数分だけのＰＵメンバー・ノードのループが配置され、各ＰＵメンバー・ノードのループ内には、ＰＵメンバー・ノードのｎｏｄｅ＿ｔａｇが書き込まれる。ＰＵメンバー・ノードは、ディレクトリーのノードとアイテムのノードを含む。

【０１８１】

また、ＰＵのループ内には、該当するＰＵにおいてプリキャッシュの対象ファイルが存在する場合はその対象ファイル・リスト（ｐｒｅ－ｃａｃｈｅ　ｉｔｅｍ）の情報を記述する。具体的には、ｎｕｍｂｅｒ＿ｏｆ＿ｐｒｅ＿ｃａｃｈｅ＿ｎｏｄｅｓはプリキャッシュの対象となるノードの数であり、ｎｕｍｂｅｒ＿ｏｆ＿ｐｒｅ＿ｃａｃｈｅ＿ｎｏｄｅｓの数分だけのプリキャッシュ・ノードのループが配置される。１つのプリキャッシュ・ノードのループ内には、プリキャッシュ・ノードを識別するｐｒｅ＿ｃａｃｈｅ＿ｎｏｄｅ＿ｔａｇが書き込まれる。

【０１８２】

ここで言うプリキャッシュの対象ファイルは、例えば、現在のデータ放送提示単位（ＰＵ）から次に参照するデータ放送提示単位（ＰＵ）を構成する放送ファイル・リストで構成される。放送番組の制作側では、このようにデータ放送のシグナリング情報でプリキャッシュの対象ファイル・リストを提示することで、受信機側では次に遷移するデータ放送提示単位に必要なファイル・データをプリキャッシュしておくことができる。その結果、放送番組に連動したタイムリーなデータ放送サービスを実現することができる。

【０１８３】

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。

【０１８４】

図２８には、方式２を実現するデータ・トランスミッション・メッセージで伝送されるデータ・コンテント・マネジメント・テーブル（ＤＣＭＴ）２８００の構成例を示している。

【０１８５】

ｔａｂｌｅ＿ｉｄ（テーブル識別）には、各種シグナリング情報においデータ・コンテント・マネジメント・テーブルであることを示す８ビットの固定値が書き込まれる。ｖｅｒｓｉｏｎ＿（バージョン）は、このデータ・コンテント・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ーである。例えばデータ・コンテント・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ーである。

【０１８６】

ｎｕｍｂｅｒ＿ｏｆ＿ｃｏｎｔｅｎｔは、パッケージに含まれるコンテントの数を示す、８ビットのパラメーターである（コンテントは、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、以下のコンテントのループが配置され、コンテント毎の情報が格納

される。
【０１８７】
１つのコンテントのループ内には、コンテントに関する情報として、ｃｏｎｔｅｎｔ＿ＩＤと、ｃｏｎｔｅｎｔ＿ｖｅｒｓｉｏｎと、ｃｏｎｔｅｎｔ＿ｃａｃｈｅ＿ｓｉｚｅと、当該コンテントに含まれるデータ放送提示単位（Ｐｒｅｓｅｎｔａｔｉｏｎ Ｕｎｉｔ：ＰＵ）に関する情報が書き込まれる。ｃｏｎｔｅｎｔ＿ＩＤは、コンテントの識別情報である。ｃｏｎｔｅｎｔ＿ｖｅｒｓｉｏｎは、コンテントのバージョンを示す。ｃｏｎｔｅｎｔ＿ｃａｃｈｅ＿ｓｉｚｅは、コンテントをキャッシュするサイズを示す。
【０１８８】
ｎｕｍｂｅｒ＿ｏｆ＿ＰＵは、コンテントに含まれるデータ放送提示単位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。
【０１８９】
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と、当該データ放送提示単位（ＰＵ）の中心となるファイル（ｐｒｉｍａｒｙ ｉｔｅｍ）を識別するＰＵ＿ｐｒｉｍａｒｙ＿ｉｔｅｍ＿ｎｏｄｅ＿ｔａｇが書き込まれる。
【０１９０】
また、ＰＵのループ内には、当該データ放送提示単位（ＰＵ）を構成する放送伝送ファイル（ｍｅｍｂｅｒ ｉｔｅｍ）リストが書き込まれる。具体的には、ｎｕｍｂｅｒ＿ｏｆ＿ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅｓは、当該データ放送提示単位（ＰＵ）に含まれる（すなわち、ＰＵのメンバーとなる）ノードの数を示す。その後に、このｎｕｍｂｅｒ＿ｏｆ＿ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅｓの数分だけのＰＵメンバー・ノードのループが配置され、各ＰＵメンバー・ノードのループ内には、ＰＵメンバー・ノードのｎｏｄｅ＿ｔａｇが書き込まれる。ＰＵメンバー・ノードは、ディレクトリーのノードとアイテムのノードを含む。
【０１９１】
また、ＰＵのループ内には、該当するＰＵにおいて、キャッシュ・メモリー４０８へのキャッシュをロックする対象となるファイル（ｌｏｃｋ ｃａｃｈｅ ｉｔｅｍ）及びロック対象のうちアンロックする対象となるファイル（ｕｎｌｏｃｋ ｃａｃｈｅ ｉｔｅｍ）の情報を記述する。具体的には、ｎｕｍｂｅｒ＿ｏｆ＿ｌｏｃｋ＿ｃａｃｈｅ＿ｎｏｄｅｓはプリキャッシュの対象となるノードの数であり、その後にｎｕｍｂｅｒ＿ｏｆ＿ｌｏｃｋ＿ｃａｃｈｅ＿ｎｏｄｅｓの数分だけのロック対象ノードのループが配置される。１つのロック対象ノードのループ内には、ロック対象ノードを識別するｌｏｃｋ＿ｃａｃｈｅ＿ｎｏｄｅ＿ｔａｇが書き込まれる。また、ｎｕｍｂｅｒ＿ｏｆ＿ｕｎｌｏｃｋ＿ｃａｃｈｅ＿ｎｏｄｅｓはプリキャッシュの対象となるノードの数であり、その後にｎｕｍｂｅｒ＿ｏｆ＿ｕｎｌｏｃｋ＿ｃａｃｈｅ＿ｎｏｄｅｓの数分だけのアンロック対象ノードのループが配置される。１つのアンロック対象ノードのループ内には、アンロック対象ノードを識別するｕｎｌｏｃｋ＿ｃａｃｈｅ＿ｎｏｄｅ＿ｔａｇが書き込まれる。
【０１９２】
ここで言うロック対象ファイルは、例えば、現在のデータ放送提示単位（ＰＵ）から次に参照するデータ放送提示単位（ＰＵ）で使用する放送ファイル・リストで構成される。放送番組の制作側では、このようにデータ放送のシグナリング情報でロック対象ファイル・リストを提示することで、受信機側では次に遷移するデータ放送提示単位に必要なファイル・データをキャッシュにロックしておくことができる。その結果、放送番組に連動したタイムリーなデータ放送サービスを実現することができる。また、アンロック対象ファイルは、例えば、現在のデータ放送提示単位（ＰＵ）では不要となる放送ファイル・リストで構成される。アンロック対象ファイルを指定することで、不要なファイルをキャッシュ・メモリー４０８から削除することができ、メモリー・サイズを節約することができる。
【０１９３】
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ｎｕ

ｍｂｅｒ＿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。

【０１９４】

要するに、データ・コンテント・マネジメント・テーブルは、１つのパッケージで各コンテント（データ放送アプリケーション）をデータ放送提示単位（ＰＵ）で管理するテーブルであり、データ・コンテント・マネジメント・テーブルを参照して、ｎｏｄｅ＿ｔａｇから、そのノードを含むデータ放送提示単位のＰＵ＿ｔａｇを取得ことができる。また、データ・コンテント・マネジメント・テーブル受信機１２側でのデータ放送用ファイルの強制キャッシュを制御するという側面を持つ。但し、データ・コンテント・マネジメント・テーブルを利用したキャッシュ制御動作の詳細については、後述に譲る。

【０１９５】

図２９には、ＭＭＴ伝送されるデータ放送アプリケーション（コンテント）の伝送、コンテントのロケーションと、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。

【０１９６】

図２９（Ａ）には、コンテントのディレクトリー構造を示している。各コンテントｃｏｎｔｅｎｔ１、２、…は、データ放送アプリケーション（ａｐｐ）とマテリアルで構成される。データ放送アプリケーションやマテリアルは、それぞれファイル・データを実体とするリソースである。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、３２ビットのｉｔｅｍ＿ＩＤで識別することができる。また、シグナリング情報内では、アイテムは１６ビットのｎｏｄｅ＿ｔａｇで識別することができる。図２９（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、コンテントの実行時（データ放送の提示時）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画像やテキストなどのモノメディア・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つのデータ放送提示単位ＰＵを構成する。図２９（Ａ）に示す例では、ｃｏｎｔｅｎｔ１は、Ａ１１．ｈｔｍｌ、Ａ１２．ｈｔｍｌ、Ａ１３．ｈｔｍｌなどの１以上のＨＴＭＬ文書をデータ放送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、コンテントの実行時に直接参照されるリソースとする。

【０１９７】

図２９（Ｂ）には、コンテントの実行時（データ放送の提示時）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、コンテント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つのデータ放送提示単位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、データ放送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）。

【０１９８】

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つのデータ放送提示単位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、データ放送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つのデータ放送提示単位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。

【０１９９】

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２９（

Ｂ）に示す例では、リソースＡ１１．ｈｔｍｌは、コンテント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、Ａ１１．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Ａ１１．ｈｔｍｌ、Ａ１２．ｈｔｍｌ、Ａ０１．ｈｔｍｌは、それぞれ１つのデータ放送提示単位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、リンクし合うデータ放送提示単位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。リソースＡ１１．ｈｔｍｌ、Ａ１２．ｈｔｍｌ、Ａ０１．ｈｔｍｌは、各々のデータ放送提示単位ＰＵにおいて中心となるファイル・データ（ｐｒｉｍａｒｙ　ｉｔｅｍ＿ｎｏｄｅ）である。

【０２００】

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ト全体で大きなリソース・グループすなわちデータ・コンテント全体を構成する。データ・コンテント全体とは、共通のｃｏｎｔｅｎｔ＿ＩＤを持つデータ放送提示単位ＰＵの範囲である。データ・コンテント・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、コンテントに含まれるすべてのデータ放送提示単位ＰＵを一括して特定することができる。図２９（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ト全体のリソース・グループ２８２０を形成している。

【０２０１】

図２９（Ｃ）には、コンテントをＭＭＴ伝送する様子を模式的に示している。コンテント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トは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。

【０２０２】

また、同じコンテントに含まれるリソースは同じａｓｓｅｔ＿ＩＤを共有し、同じＭＭＴ伝送路上で伝送される。図２９（Ｃ）に示す例では、Ｉｔｅｍ＿ＩＤがｉ１１、ｉ１２、ｉ１３、ｉ１４の各アイテムは、同じＡｓｓｅｔ＿ＩＤとしてａ１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・ディレクトリー・マネジメント・テーブルは図２９（Ａ）で表現され、データ・コンテント・マネジメント・テーブルは図２９（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２９（Ｃ）で表現され、これらの間をｉｔｅｍ＿ＩＤ若しくはｎｏｄｅ＿ｔａｇにより関係付けられることになる。

【０２０３】

ＭＭＴ伝送路からデータ放送アプリケーション（コンテント）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３０を参照しながら説明する。

【０２０４】

受信機１２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＿ｐｒｏｔｏ

ｃｏｌ＿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・ディレクトリー・マネジメント・テーブル（ＤＤＭＴ）２９０２を参照して、そのｂａｓｅ＿ｆｏｌｄｅｒ＿ｐａｔｈ＿ｂｙｔｅ、ｆｏｌｄｅｒ＿ｎｏｄｅ＿ｐａｔｈ＿ｂｙｔｅ、及びｆｉｌｅ＿ｎａｍｅ＿ｂｙｔｅの組み合わせに対応するアイテムのｎｏｄｅ＿ｔａｇを取得することができる。

【０２０５】

次いで、受信機１２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ｎｏｄｅ＿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。

【０２０６】

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。

【０２０７】

また、受信機１２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・コンテント・マネジメント・テーブル（ＤＣＭＴ）２９０５内で、データ・ディレクトリー・マネジメント・テーブル２９０２から取得したｎｏｄｅ＿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。

【０２０８】

図３１には、受信機内で、データ放送に利用されるファイル・データをキャッシュする仕組みを模式的に示している。ここで言うキャッシュには、実行するファイル・データをキャッシュすることとプリキャッシュすることを含む。

【０２０９】

アプリケーション・データ制御部４０７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。コンテントのプリキャッシュに関しては、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブルに含まれている強制キャッシュ情報に基づいて、放送番組に連動したデータ放送で利用される各ファイルをプリキャッシュする。

【０２１０】

具体的には、アプリケーション・データ制御部４０７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・コンテント・マネジメント・テーブルでプリキャッシュすることが指定されたノード（ディレクトリー、又はファイル・データ）のｃａｃｈｅ＿ｎｏｄｅ＿ｔａｇを取得する。ｎｏｄｅ＿ｔａｇから該当するＭＭＴＰパケットを特定できることは、図３０を参照しながら説明した通りである。

【０２１１】

アプリケーション・データ制御部４０７は、プリキャッシュしたいファイル・データのｎｏｄｅ＿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、そのノード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ーを取得して、キャッシュ・メモリー４０８にプリキャッシュする。

【０２１２】

データ放送アプリケーション・エンジン４０９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）が既にキャッシュ・メモリー４０８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、キャッシュ・メモリー４０８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。一方、必要なアイテムがキャッシュ・メモリー４０８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。

【０２１３】

続いて、受信機１２において、データ放送で利用されるファイル・データをキャッシュする制御動作について詳解する。

【０２１４】

データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）による強制キャッシュ情報の伝送方式として、方式１及び方式２の２通りがあることは既に述べた。まず方式１を利用したキャッシュ制御動作について説明する。

【０２１５】

方式１では、放送送出システム１１側からは、図２７に示したデータ・コンテント・マネジメント・テーブル（ＤＣＭＴ）２７００がデータ・トランスミッション・メッセージで伝送される。データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）２７００は、データ放送提示単位（ＰＵ）毎に、提示単位を構成する放送伝送ファイル（ｍｅｍｂｅｒ　ｉｔｅｍ）リストと中心となるファイル（ｐｒｉｍａｒｙ　ｉｔｅｍ）、及びプリキャッシュの対象ファイルが存在する場合はその対象ファイル・リスト（ｐｒｅ－ｃａｃｈｅ　ｉｔｅｍ）の情報を記述する。したがって、受信機１２側では、以下に示すような、プリキャッシュによるキャッシュ制御動作を行なうことができる。

【０２１６】

（動作０１）アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０４上で伝送されるデータ・トランスミッション・メッセージを適宜検出して更新を行ないつつ、最新の情報を取得する。

（動作０２）アプリケーション・データ制御部４０７が、データ放送アプリケーション制御エンジン４０９などからの指示によりｐｒｉｍａｒｙ＿ｉｔｅｍに指定された（データ放送提示単位の中心となる）アプリケーション・ファイルにアクセスすることにより、対応するデータ放送提示単位（ＰＵ）の提示状態に入ったことを認識する。

（動作０３）アプリケーション・データ制御部４０７は、データ・トランスミッション・メッセージに含まれるデータ・コンテント・マネジメント・テーブル（ＤＣＭＴ）において、該当するデータ放送提示単位（ＰＵ）に含まれる各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）も同時に取得して、キャッシュ・メモリー４０８に保持する。

（動作０４）さらに、データ・コンテント・マネジメント・テーブル内で、該当するデータ放送提示単位（ＰＵ）に対してプリキャッシュの対象が指定されている場合には、アプリケーション・データ制御部４０７は、プリキャッシュ対象の各ファイル（ｉｔｅｍ）も取得して、キャッシュ・メモリー４０８にプリキャッシュする。

（動作０５）データ放送アプリケーション・エンジン４０９は、ｐｒｉｍａｒｙ＿ｉｔｅｍに指定されたアプリケーション・ファイルを、キャッシュ・メモリー４０８から実行する。

（動作０６）その後、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）の更新

により、現在提示状態にあるデータ放送提示単位（ＰＵ）に含まれるメンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）やプリキャッシュ対象のファイルの構成が変化した場合、又は、データ放送アプリケーション・エンジン４０９におけるアプリケーション動作により他のデータ放送提示単位（ＰＵ）の提示状態に遷移した場合などには、現在提示状態にあるデータ放送提示単位（ＰＵ）において、上記の（動作０３）～（動作０５）の処理を行なう。以前の状態でファイルをキャッシュ・メモリー４０８に保持していても、アプリケーション・データ制御部４０７は、現在の状態で不要なファイルをキャッシュ・メモリー４０８から削除する。

【０２１７】

図３２には、受信機１２における方式１に基づくファイル・データのキャッシュ制御手順をフローチャートの形式で示している。

【０２１８】

データ放送アプリケーション制御エンジン４０９がアプリケーション動作すなわちデータ放送の提示を行なっているときに（ステップＳ３２０１のＹｅｓ）、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブル内でｐｒｉｍａｒｙ＿ｉｔｅｍ（データ放送提示単位の中心）に指定されたアプリケーション・ファイルにアクセスすることにより（ステップＳ３２０２のＹｅｓ）、対応するデータ放送提示単位（ＰＵ）の提示状態に入ったことを認識する。

【０２１９】

アプリケーション・データ制御部４０７は、キャッシュ・メモリー４０８をリセットして、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）において、現在提示しているデータ放送提示単位（ＰＵ）のｐｒｉｍａｒｙ＿ｉｔｅｍ並びに各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）を取得して、キャッシュ・メモリー４０８に保持する（ステップＳ３２０３）。データ放送アプリケーション・エンジン４０９は、ｐｒｉｍａｒｙ＿ｉｔｅｍに指定されたアプリケーション・ファイルを、キャッシュ・メモリー４０８から実行する。

【０２２０】

また、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブル内で、該当するデータ放送提示単位（ＰＵ）に対してプリキャッシュの対象が指定されているかどうかをチェックする（ステップＳ３２０４）。そして、プリキャッシュの対象が指定されている場合には（ステップＳ３２０４のＹｅｓ）、アプリケーション・データ制御部４０７は、プリキャッシュ対象の各ファイル（ｉｔｅｍ）も取得して、キャッシュ・メモリー４０８にプリキャッシュする（ステップＳ３２０５）。

【０２２１】

その後、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）の更新により、現在提示状態にあるデータ放送提示単位（ＰＵ）に含まれるメンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）やプリキャッシュ対象のファイルの構成が変化した場合には（ステップＳ３２０６のＹｅｓ）、ステップＳ３２０３に戻り、現在提示状態にあるデータ放送提示単位（ＰＵ）において、上記の処理を繰り返し実行する。

【０２２２】

また、データ放送アプリケーション・エンジン４０９におけるアプリケーション動作により他のデータ放送提示単位（ＰＵ）の提示状態に遷移し、そのデータ放送提示単位のｐｒｉｍａｒｙ＿ｉｔｅｍに指定されたアプリケーション・ファイルにアクセスした場合には（ステップＳ３２０７のＹｅｓ）、ステップＳ３２０３に戻り、遷移した先のデータ放送提示単位（ＰＵ）において、上記の処理を繰り返し実行する。

【０２２３】

以上の処理を、データ放送アプリケーション・エンジン４０９がアプリケーション動作を終了するまで（ステップＳ３２０８のＮｏ）、繰り返し実行する。

【０２２４】

図３３には、受信機１２における方式１に基づくファイル・データのプリキャッシュ動

作例を示している。

【０２２５】

アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０４上で受信する各種シグナリング・メッセージを解析している。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・メッセージには、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）が含まれている。

【０２２６】

データ放送アプリケーション制御エンジン４０９がＰＵ＿ｉｄ＝１で識別されるデータ放送提示単位（ＰＵ）のアプリケーション動作を行なっているとき、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブルを参照して、現在提示状態にあるデータ放送提示単位（ＰＵ＿ｉｄ＝１）のｐｒｉｍａｒｙ＿ｉｔｅｍ（データ放送提示単位の中心）に指定されたアプリケーション・ファイル「Ａ０１．ｈｔｍｌ」、並びに、各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）「Ｂ０１」、「Ｂ０２」それぞれのｎｏｄｅ＿ｔａｇをデータ・コンテント・マネジメント・テーブルから取得して、データ・アセットを伝送するＭＭＴ伝送路５０３からこれらのｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得すると、参照番号３３０１で示すように、キャッシュ・メモリー４０８にキャッシュする。なお、ｎｏｄｅ＿ｔａｇからＭＭＴ伝送路５０３上で伝送されるファイルにアクセスする方法については、図３０を参照しながら、既に説明した通りである（以下、同様）。但し、このデータ放送提示単位（ＰＵ）に対してプリキャッシュの対象が指定されていないので、プリキャッシュ動作は行なわない。

【０２２７】

次いで、データ放送アプリケーション・エンジン４０９におけるアプリケーション動作により、ＰＵ＿ｉｄ＝３で識別される他のデータ放送提示単位（ＰＵ）の提示状態に遷移したとする。アプリケーション・データ制御部４０７は、上記と同様に、遷移した先のデータ放送提示単位（ＰＵ＿ｉｄ＝３）のｐｒｉｍａｒｙ＿ｉｔｅｍ（データ放送提示単位の中心）に指定されたアプリケーション・ファイル「Ａ１１．ｈｔｍｌ」、並びに、各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）「Ｂ１１」、「Ｂ１２」、「Ｂ１３」それぞれのｎｏｄｅ＿ｔａｇをデータ・コンテント・マネジメント・テーブルから取得すると、ＭＭＴ伝送路５０３からこれらのｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得して、参照番号３３０２で示すように、キャッシュ・メモリー４０８にキャッシュする。但し、このデータ放送提示単位（ＰＵ）に対してプリキャッシュの対象が指定されていないので、プリキャッシュ動作は行なわない。

【０２２８】

アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０４上で受信する各種シグナリング・メッセージを常に解析している。そして、参照番号３３０３で示すように、データ・コンテント・マネジメント・テーブルのバージョンが１から２に更新されたことを検出すると、アプリケーション・データ制御部４０７は、現在提示状態にあるデータ放送提示単位（ＰＵ＿ｉｄ＝３）のｐｒｉｍａｒｙ＿ｉｔｅｍ並びに各メンバー・ファイル、プリキャッシュ対象のファイルに変更がないかどうかをチェックする。今回はメンバー・ファイルに変更がないので、以前キャッシュしておいた各メンバー・ファイルはキャッシュ・メモリー４０８に保持したままとする。また、データ放送提示単位（ＰＵ＿ｉｄ＝３）のプリキャッシュ対象のファイルとして「Ａ１２」、「Ｂ１４」、「Ｂ１５」が追加されているので、アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０３からこれらのｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得して、参照番号３３０４で示すように、キャッシュ・メモリー４０８にキャッシュする。

【０２２９】

次いで、参照番号３３０５で示すように、データ放送の提示の更新を指示するイベント・メッセージをＭＭＴ伝送路５０４から受信すると、データ放送アプリケーション・エンジン４０９は、実行中のＡ１１ファイルからＡ１２ファイルへのＨＴＭＬ文書の遷移を行

なう。一方でほぼ同時に、アプリケーション・データ制御部４０７は、バージョンが２から３に更新されたデータ・コンテント・マネジメント・テーブルを参照して、現在提示状態にあるデータ放送提示単位（ＰＵ＿ｉｄ＝３）のｐｒｉｍａｒｙ＿ｉｔｅｍが「Ａ１２．ｈｔｍｌ」に変更するとともに、メンバー・ファイルが「Ｂ１４」、「Ｂ１５」、「Ｂ１６」に変更したことを検出する。上述したように、ｐｒｉｍａｒｙ＿ｉｔｅｍ「Ａ１２．ｈｔｍｌ」並びにメンバー・ファイルが「Ｂ１４」、「Ｂ１５」は既にキャッシュ・メモリー４０８にプリキャッシュされているので、データ放送アプリケーション・エンジン４０９は、キャッシュ・メモリー４０８に保持されているファイルを利用してデータ放送を迅速に表示することができる。すなわち、放送番組に連動したタイムリーなデータ放送の提示を行なうことができる。また、データ放送提示単位（ＰＵ＿ｉｄ＝３）のプリキャッシュ対象のファイルとして「Ｂ１２」が追加されているので、アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０３からそのｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得して、参照番号３３０６で示すように、キャッシュ・メモリー４０８にキャッシュする。

【０２３０】

続いて、方式２を利用したキャッシュ制御動作について説明する。

【０２３１】

方式２では、放送送出システム１１側からは、図２８に示したデータ・コンテント・マネジメント・テーブル（ＤＣＭＴ）２８００がデータ・トランスミッション・メッセージで伝送される。データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）２８００は、データ放送提示単位（ＰＵ）毎に、提示単位を構成する放送伝送ファイル（ｍｅｍｂｅｒ　ｉｔｅｍ）リストと中心となるファイル（ｐｒｉｍａｒｙ　ｉｔｅｍ）、及びキャッシュにロックする対象ファイル（ｌｏｃｋ　ｃａｃｈｅ　ｉｔｅｍ）及びロック対象のうちアンロックする対象ファイル（ｕｎｌｏｃｋ　ｃａｃｈｅ　ｉｔｅｍ）の情報を記述する。したがって、受信機１２側では、以下に示すような、キャッシュのロック並びにアンロックによるキャッシュ制御動作を行なうことができる。

【０２３２】

（動作１１）アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０４上で伝送されるデータ・トランスミッション・メッセージを適宜検出して更新を行ないつつ、最新の情報を取得する。

（動作１２）アプリケーション・データ制御部４０７が、データ放送アプリケーション制御エンジン４０９などからの指示によりｐｒｉｍａｒｙ＿ｉｔｅｍに指定された（データ放送提示単位の中心となる）アプリケーション・ファイルにアクセスすることにより、対応するデータ放送提示単位（ＰＵ）の提示状態に入ったことを認識する。

（動作１３）アプリケーション・データ制御部４０７は、データ・トランスミッション・メッセージに含まれるデータ・コンテント・マネジメント・テーブル（ＤＣＭＴ）において、該当するデータ放送提示単位（ＰＵ）に含まれる各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）も同時に取得して、キャッシュ・メモリー４０８に保持する。

（動作１４）さらに、データ・コンテント・マネジメント・テーブル内で、該当するデータ放送提示単位（ＰＵ）に対してロックキャッシュの対象が指定されている場合には、アプリケーション・データ制御部４０７は、ロックキャッシュ対象のうち未取得の各ファイル（ｉｔｅｍ）も取得して、キャッシュ・メモリー４０８にキャッシュし、且つ、ロックキャッシュ対象ファイルとして別途管理する。

（動作１５）逆に、データ・コンテント・マネジメント・テーブル内で、該当するデータ放送提示単位（ＰＵ）に対してアンロックキャッシュの対象に指定されている場合には、アプリケーション・データ制御部４０７は、アンロック対象ファイルがキャッシュ・メモリー４０８にキャッシュされていれば、これを削除するとともに、別途管理していたロックキャッシュ対象ファイルからも削除する。

（動作１６）データ放送アプリケーション・エンジン４０９は、ｐｒｉｍａｒｙ＿ｉｔｅｍに指定されたアプリケーション・ファイルを、キャッシュ・メモリー４０８から実行す

る。
（動作１７）その後、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）の更新により、現在提示状態にあるデータ放送提示単位（ＰＵ）に含まれるメンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）やプリキャッシュ対象のファイルの構成が変化した場合には、上記の（動作１３）～（動作１６）の処理を行なう。
（動作１８）データ放送アプリケーション・エンジン４０９におけるアプリケーション動作により他のデータ放送提示単位（ＰＵ）の提示状態に遷移した場合には、ロックキャッシュ対象のファイルは一旦キャッシュ・メモリー４０８に保持した上で、上記の（動作１３）～（動作１６）の処理を行なう。ロックキャッシュ対象以外のファイルは、キャッシュ・メモリー４０８から削除してもよい。

【０２３３】
図３４には、受信機１２における方式２に基づくファイル・データのキャッシュ制御手順をフローチャートの形式で示している。

【０２３４】
データ放送アプリケーション制御エンジン４０９がアプリケーション動作すなわちデータ放送の提示を行なっているときに（ステップＳ３４０１のＹｅｓ）、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブル内でｐｒｉｍａｒｙ＿ｉｔｅｍ（データ放送提示単位の中心）に指定されたアプリケーション・ファイルにアクセスすることにより（ステップＳ３４０２のＹｅｓ）、対応するデータ放送提示単位（ＰＵ）の提示状態に入ったことを認識する。

【０２３５】
アプリケーション・データ制御部４０７は、キャッシュ・メモリー４０８をリセットして、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）において、現在提示しているデータ放送提示単位（ＰＵ）のｐｒｉｍａｒｙ＿ｉｔｅｍ並びに各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）を取得して、キャッシュ・メモリー４０８に保持する（ステップＳ３４０３）。データ放送アプリケーション・エンジン４０９は、ｐｒｉｍａｒｙ＿ｉｔｅｍに指定されたアプリケーション・ファイルを、キャッシュ・メモリー４０８から実行する。

【０２３６】
また、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブル内で、該当するデータ放送提示単位（ＰＵ）に対してロックキャッシュの対象が指定されているかどうかをチェックする（ステップＳ３４０４）。そして、ロックキャッシュの対象が指定されている場合には（ステップＳ３４０４のＹｅｓ）、アプリケーション・データ制御部４０７は、ロックキャッシュ対象のうち未取得の各ファイル（ｉｔｅｍ）も取得して、キャッシュ・メモリー４０８にキャッシュし、且つ、ロックキャッシュ対象ファイルとして別途管理する（ステップＳ３４０５）。

【０２３７】
また、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブル内で、該当するデータ放送提示単位（ＰＵ）に対してアンロックキャッシュの対象が指定されているかどうかをチェックする（ステップＳ３４０６）。そして、アンロックキャッシュの対象が指定されている場合には（ステップＳ３４０６のＹｅｓ）、アプリケーション・データ制御部４０７は、アンロック対象ファイルがキャッシュ・メモリー４０８にキャッシュされていれば、これを削除するとともに、別途管理していたロックキャッシュ対象ファイルからも削除する（ステップＳ３４０７）。

【０２３８】
その後、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）の更新により、現在提示状態にあるデータ放送提示単位（ＰＵ）に含まれるメンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）やプリキャッシュ対象のファイルの構成が変化した場合には（ステップＳ３４０８のＹｅｓ）、ステップＳ３４０３に戻り、現在提示状態にあるデータ放送提示単位（ＰＵ）において、上記の処理を繰り返し実行する。

【０２３９】
　また、データ放送アプリケーション・エンジン４０９におけるアプリケーション動作により他のデータ放送提示単位（ＰＵ）の提示状態に遷移し、そのデータ放送提示単位のｐｒｉｍａｒｙ＿ｉｔｅｍに指定されたアプリケーション・ファイルにアクセスした場合には（ステップＳ３４０９のＹｅｓ）、ステップＳ３４０３に戻り、遷移した先のデータ放送提示単位（ＰＵ）において、上記の処理を繰り返し実行する。
【０２４０】
　以上の処理を、データ放送アプリケーション・エンジン４０９がアプリケーション動作を終了するまで（ステップＳ３４１０のｎｏ）、繰り返し実行する。
【０２４１】
　図３５には、受信機１２における方式２に基づくファイル・データのキャッシュのロック及びアンロック動作例を示している。
【０２４２】
　アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０４上で受信する各種シグナリング・メッセージを解析している。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・メッセージには、データ・コンテント・マネジメント・テーブル（ＤＣＭＴ）が含まれている。
【０２４３】
　データ放送アプリケーション制御エンジン４０９がＰＵ＿ｉｄ＝１で識別されるデータ放送提示単位（ＰＵ）のアプリケーション動作を行なっているとき、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブルを参照して、現在提示状態にあるデータ放送提示単位（ＰＵ＿ｉｄ＝１）のｐｒｉｍａｒｙ＿ｉｔｅｍ（データ放送提示単位の中心）に指定されたアプリケーション・ファイル「Ａ０１．ｈｔｍｌ」、並びに、各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）「Ｂ０１」、「Ｂ０２」それぞれのｎｏｄｅ＿ｔａｇをデータ・コンテント・マネジメント・テーブルから取得して、データ・アセットを伝送するＭＭＴ伝送路５０３からこれらのｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得すると、参照番号３５０１で示すように、キャッシュ・メモリー４０８にキャッシュする。なお、ｎｏｄｅ＿ｔａｇからＭＭＴ伝送路５０３上で伝送されるファイルにアクセスする方法については、図３０を参照しながら、既に説明した通りである（以下、同様）。但し、このデータ放送提示単位（ＰＵ）に対してロックキャッシュの対象が指定されていないので、キャッシュ動作は行なわない。
【０２４４】
　次いで、データ放送アプリケーション・エンジン４０９におけるアプリケーション動作により、ＰＵ＿ｉｄ＝３で識別される他のデータ放送提示単位（ＰＵ）の提示状態に遷移したとする。アプリケーション・データ制御部４０７は、上記と同様に、遷移した先のデータ放送提示単位（ＰＵ＿ｉｄ＝３）のｐｒｉｍａｒｙ＿ｉｔｅｍ（データ放送提示単位の中心）に指定されたアプリケーション・ファイル「Ａ１１．ｈｔｍｌ」、並びに、各メンバー・ファイル（ＰＵ＿ｍｅｍｂｅｒ＿ｎｏｄｅ）「Ｂ１１」、「Ｂ１２」、「Ｂ１３」それぞれのｎｏｄｅ＿ｔａｇをデータ・コンテント・マネジメント・テーブルから取得すると、ＭＭＴ伝送路５０３からこれらのｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得して、参照番号３５０２で示すように、キャッシュ・メモリー４０８にキャッシュする。但し、このデータ放送提示単位（ＰＵ）に対してロックキャッシュの対象が指定されていないので、ロックキャッシュ動作は行なわない。
【０２４５】
　アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０４上で受信する各種シグナリング・メッセージを常に解析している。そして、参照番号３５０３で示すように、データ・コンテント・マネジメント・テーブルのバージョンが１から２に更新されたことを検出すると、アプリケーション・データ制御部４０７は、現在提示状態にあるデータ放送提示単位（ＰＵ＿ｉｄ＝３）のｐｒｉｍａｒｙ＿ｉｔｅｍ並びに各メンバー・ファイル、

ロックキャッシュ対象のファイルに変更がないかどうかをチェックする。今回はメンバー・ファイルに変更がないので、以前キャッシュしておいた各メンバー・ファイルはキャッシュ・メモリー４０８に保持したままとする。また、データ放送提示単位（ＰＵ＿ｉｄ＝３）のロックキャッシュ対象のファイルとして「Ａ１２」、「Ｂ１４」、「Ｂ１２」、「Ｂ１５」が追加されているので、アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０３からこれらのｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得して、参照番号３５０４で示すように、キャッシュ・メモリー４０８にキャッシュするとともに、ロックキャッシュ対象ファイルとして別途管理する。図中、ロックキャッシュ対象として管理されているファイルを下線で示している（以下、同様）。

【０２４６】

次いで、参照番号３５０５で示すように、データ放送の提示の更新を指示するイベント・メッセージをＭＭＴ伝送路５０４から受信すると、データ放送アプリケーション・エンジン４０９は、実行中のＡ１１ファイルからＡ１２ファイルへのＨＴＭＬ文書の遷移を行なう。一方でほぼ同時に、アプリケーション・データ制御部４０７は、バージョンが２から３に更新されたデータ・コンテント・マネジメント・テーブルを参照して、現在提示状態にあるデータ放送提示単位（ＰＵ＿ｉｄ＝３）のｐｒｉｍａｒｙ＿ｉｔｅｍが「Ａ１２．ｈｔｍｌ」に変更するとともに、メンバー・ファイルが「Ｂ１４」、「Ｂ１５」、「Ｂ１６」に変更したことを検出する。上述したように、ｐｒｉｍａｒｙ＿ｉｔｅｍ「Ａ１２．ｈｔｍｌ」並びにメンバー・ファイルが「Ｂ１４」、「Ｂ１５」は既にキャッシュ・メモリー４０８にロックキャッシュされているので、データ放送アプリケーション・エンジン４０９は、キャッシュ・メモリー４０８に保持されているファイルを利用してデータ放送を迅速に表示することができる。すなわち、放送番組に連動したタイムリーなデータ放送の提示を行なうことができる。また、このデータ放送提示単位（ＰＵ）に対してロックキャッシュの対象が指定されていないので、ロックキャッシュ動作は行なわない。参照番号３５０６で示すように、ロックキャッシュされたファイル「Ａ１２」、「Ｂ１４」、「Ｂ１２」、「Ｂ１５」がそのままキャッシュ・メモリー４０８に保持される一方、不要になったファイル「Ａ１１」が削除されている。

【０２４７】

さらに、データ放送の提示の更新を指示するイベント・メッセージをＭＭＴ伝送路５０４から受信すると、アプリケーション・データ制御部４０７は、データ・コンテント・マネジメント・テーブルを参照して、現在提示状態にあるデータ放送提示単位（ＰＵ＿ｉｄ＝３）に対して、ロックキャッシュ対象のファイルとして「Ｂ１７」が追加されているとともに、アンロック対象ファイルとして「Ｂ１４」が追加されている。そこで、参照番号３５０７で示すように、アプリケーション・データ制御部４０７は、ＭＭＴ伝送路５０３からファイル「Ｂ１７」のｎｏｄｅ＿ｔａｇに対応するファイル・データのエンティティーを取得してキャッシュ・メモリー４０８にロックキャッシュするとともに、ロックキャッシュしていたファイル「Ｂ１２」をキャッシュ・メモリー４０８から削除しロック対象から外す。

【０２４８】

現在運用されているＢＭＬ（Ｂｒｏａｄｃａｓｔ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ）によるデータ放送サービスでは、スクリプトから「ＬｏｃｋＭｏｄｕｌｅＯｎＭｅｍｏｒｙ（）」というＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ Ｐｒｏｇｒａｍｍｉｎｇ Ｉｎｔｅｒｆａｃｅ）を呼び出すことにより、特定のファイルをあらかじめキャッシュ・メモリーにプリキャッシュして留めておくことが可能である（例えば、特許文献２を参照のこと）。この方法は、スクリプトなどのアプリケーションの仕様に、放送運用を前提とする特殊な仕様を盛り込む必要がある。

【０２４９】

これに対し、本明細書で開示する技術によれば、放送局などの送信側からは、データ放送に関わるシグナリングに、強制キャッシュを指定する情報を含めて伝送し、受信機側では、受信したデータ放送に関わるシグナリングに含まれている強制キャッシュ情報に基づ

いてデータ放送用の各ファイルのキャッシュ制御を行なうようになっている。したがって、本明細書で開示する技術によれば、新しいＨＴＭＬ５によるデータ放送において、スクリプトなどのアプリケーションの仕様に、放送運用を前提とする特殊な仕様を盛り込むことなく、汎用性の高いフォーマットを維持することができる。

【産業上の利用可能性】

【０２５０】

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。

【０２５１】

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、放送番組に連動するデータ放送に利用されるファイル・データをＭＭＴ方式又はその他のトランスポート方式により伝送するさまざまなデータ放送システムに適用することができる。

【０２５２】

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。

【０２５３】

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）データ放送で利用されるファイル・データを送信するファイル・データ送信部と、

データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ送信部と、

を具備する送信装置。

（２）前記シグナリング・メッセージ送信部は、データ放送提示単位毎に、提示単位を構成する放送伝送ファイル・リストと中心となるファイル、及び、プリキャッシュの対象ファイル・リストの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを送信する、

上記（１）に記載の送信装置。

（３）前記シグナリング・メッセージ送信部は、データ放送提示単位毎に、提示単位を構成する放送伝送ファイル・リストと中心となるファイル、キャッシュにロックする対象ファイル及びロック対象のうちアンロックする対象ファイルの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを送信する、

上記（１）に記載の送信装置。

（４）データ放送が連動する放送番組本体のメディア・データを送信するメディア・データ送信部をさらに備える、

上記（１）乃至（３）のいずれかに記載の送信装置。

（５）データ放送で利用されるファイル・データを送信するファイル・データ送信ステップと、

データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ送信ステップと、

を有する送信方法。

（６）データ放送で利用されるファイル・データを送信するファイル・データ受信部と、

データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ受信部と、

前記強制キャッシュ情報に基づいて前記ファイル・データ受信部が受信するファイル・データのキャッシュ・メモリーへのキャッシュを制御する制御部と、

を具備する受信装置。

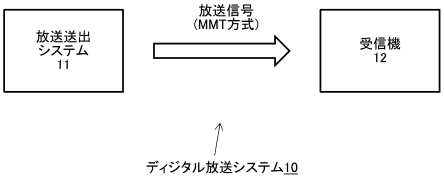
（７）前記シグナリング・メッセージ受信部は、データ放送提示単位毎に、提示単位を構

成する放送伝送ファイル・リストと中心となるファイル、及び、プリキャッシュの対象ファイル・リストの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを受信する、
上記（６）に記載の受信装置。
（８）前記制御部は、前記プリキャッシュの対象ファイル・リストに含まれているファイルを前記ファイル・データが受信すると、前記キャッシュ・メモリーにプリキャッシュする、
上記（７）に記載の受信装置。
（９）前記シグナリング・メッセージ受信部は、データ放送提示単位毎に、提示単位を構成する放送伝送ファイル・リストと中心となるファイル、キャッシュにロックする対象ファイル及びロック対象のうちアンロックする対象ファイルの情報を記述したデータ・コンテント・マネジメント・テーブルを含んだデータ・トランスミッション・メッセージを受信する、
上記（６）に記載の受信装置。
（１０）前記制御部は、前記ロック対象のファイルを前記ファイル・データが受信すると、前記キャッシュ・メモリーにプリキャッシュする、
上記（９）に記載の受信装置。
（１１）前記制御部は、前記アンロック対象のファイルを前記キャッシュ・メモリーから削除する、
上記（９）に記載の受信装置。
（１２）前記制御部は、現在のデータ放送提示単位を構成する放送伝送ファイル・リストと中心となるファイルを前記ファイル・データが受信すると、前記キャッシュ・メモリーにキャッシュする、
上記（６）又は（９）のいずれかに記載の受信装置。
（１３）ファイル・データを利用してデータ放送を提示するデータ放送提示部をさらに備える、
上記（６）乃至（１２）のいずれかに記載の受信装置。
（１４）データ放送が連動する放送番組本体のメディア・データを受信するメディア・データ受信部と、メディア・データに基づいて放送番組を提示する放送番組提示部をさらに備える、
上記（６）乃至（１３）のいずれかに記載の受信装置。
（１５）データ放送で利用されるファイル・データを送信するファイル・データ受信ステップと、
　データ放送に関わるシグナリングにファイル・データの強制キャッシュを指定する強制キャッシュ情報を含めて送信するシグナリング・メッセージ受信ステップと、
　前記強制キャッシュ情報に基づいて前記ファイル・データ受信部が受信するファイル・データのキャッシュ・メモリーへ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。
【符号の説明】
【０２５４】
　１０…ディ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
　４０１…チューナー・復調部、４０２…デマルチプレクサー
　４０３…時計部、４０４…ビデオ・デコーダー

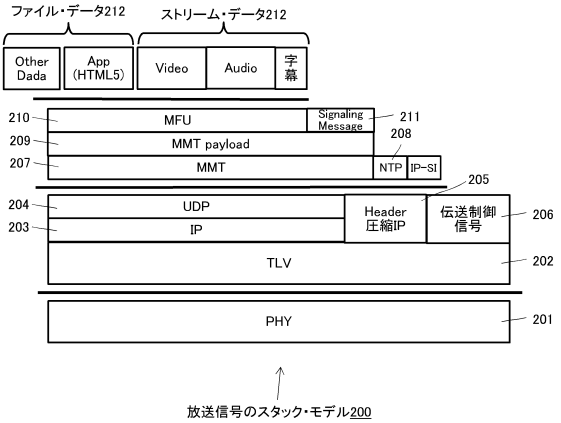
４０５…オーディオ・デコーダー、４０６…キャプション・デコーダー
４０７…アプリケーション・データ制御部、４０８…キャッシュ・メモリー
４０９…データ放送アプリケーション・エンジン
４１０…システム制御部、４１１…合成部
４１２…ＩＰインターフェース

【図面】

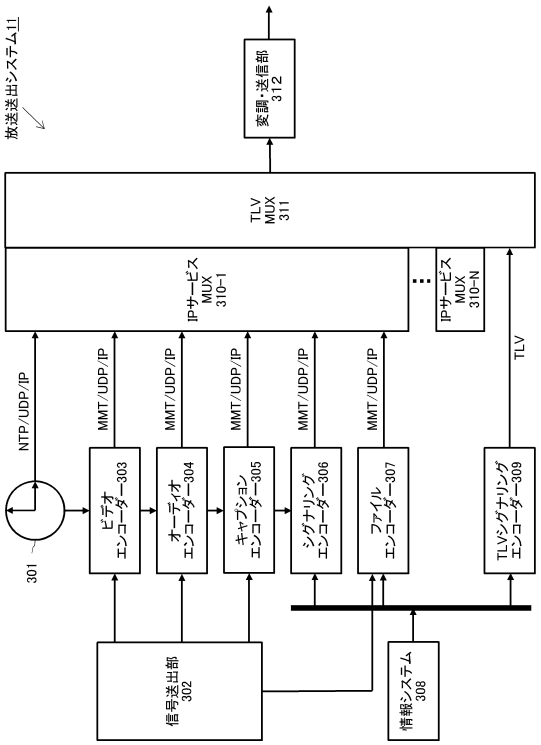
【図１】



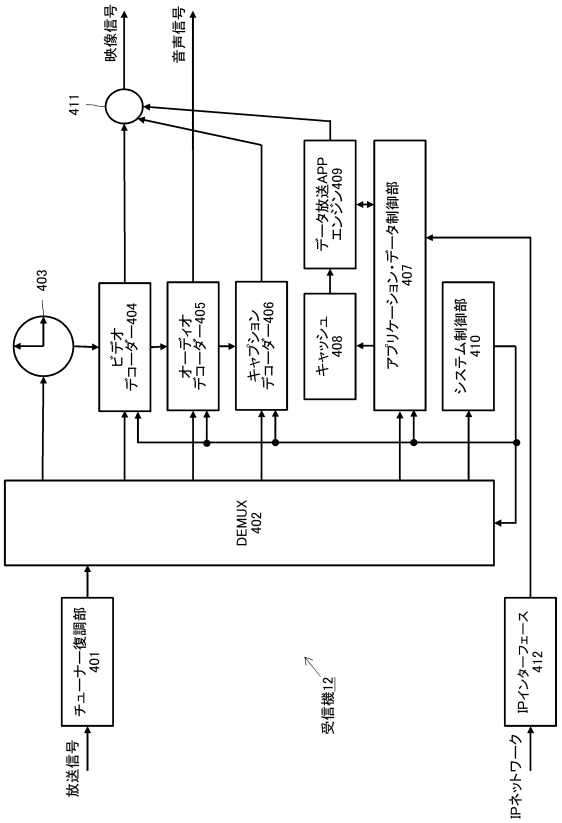
【図２】



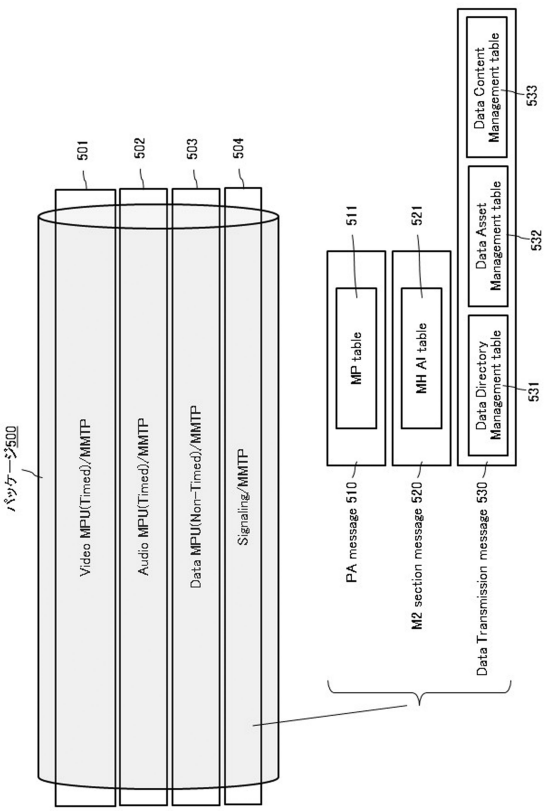
【図３】


放送送出システム11
変調・送信部 312
TLV MUX 311
IPサービス MUX 310-1
IPサービス MUX 310-N
NTP/UDP/IP
MMT/UDP/IP
TLV
301
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
シグナリング エンコーダー306
ファイル エンコーダー307
TLVシグナリング エンコーダー309
信号送出部 302
情報システム 308

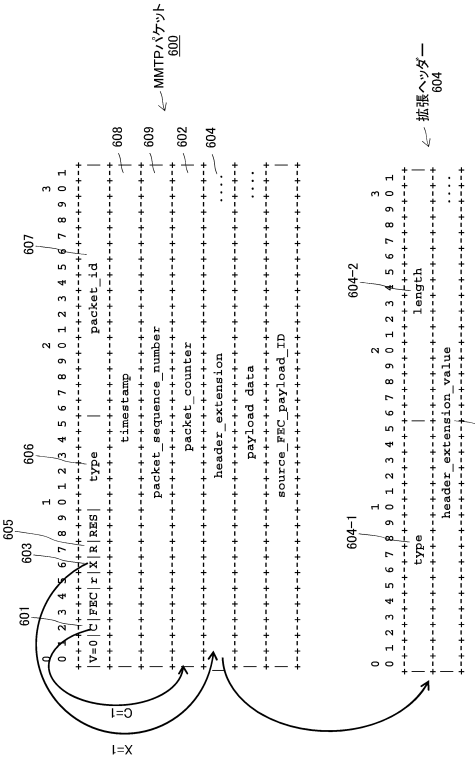
【図４】


映像信号
音声信号
411
403
ビデオ デコーダー404
オーディオ デコーダー405
キャプション デコーダー406
キャッシュ 408
データ放送APP エンジン409
アプリケーション・データ制御部 407
システム制御部 410
DEMUX 402
チューナー復調部 401
IPインターフェース 412
放送信号
IPネットワーク
受信機12

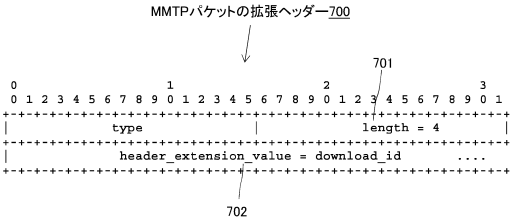
【図５】


パッケージ500
501
502
503
504
Video MPU(Timed)/MMTP
Audio MPU(Timed)/MMTP
Data MPU(Non-Timed)/MMTP
Signaling/MMTP
511
521
MP table
MH AI table
Data Directory Management table
Data Asset Management table
Data Content Management table
531
532
533
PA message 510
M2 section message 520
Data Transmission message 530

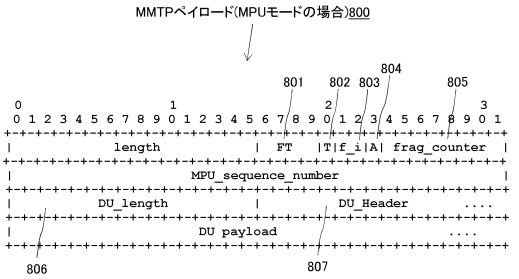
【図６】


MMTPパケット 600
拡張ヘッダー 604
601
602
603
604
604-1
604-2
604-3
605
606
607
608
609
X=1
0=1

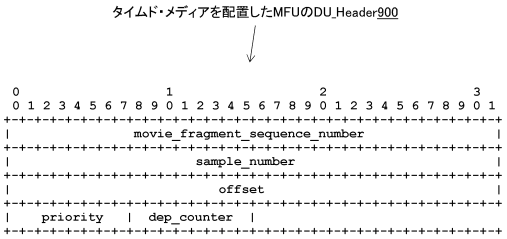
【図７】


MMTPパケットの拡張ヘッダー700
701
type
length = 4
header_extension_value = download_id
....
702

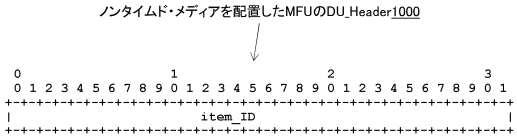
【図８】


MMTPペイロード(MPUモードの場合)800
801
802
803
804
805
length
FT
T
f_i
A
frag_counter
MPU_sequence_number
DU_length
DU_Header
....
DU payload
806
807

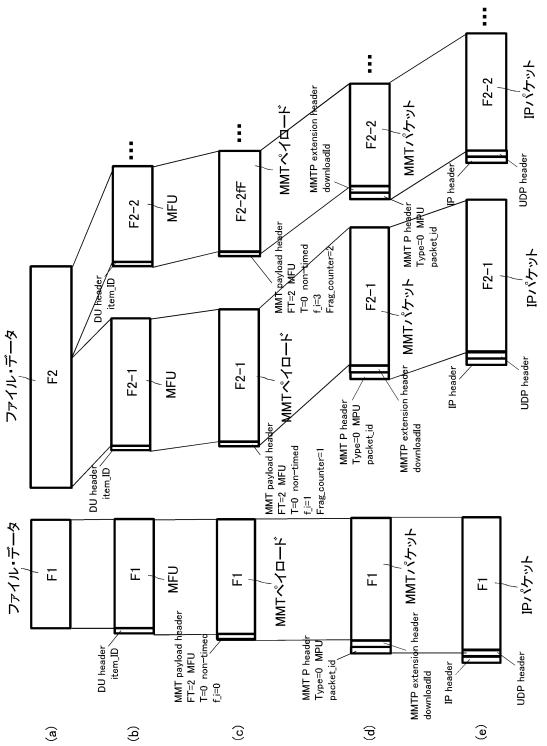
【図９】


タイムド・メディアを配置したMFUのDU_Header900
movie_fragment_sequence_number
sample_number
offset
priority
dep_counter

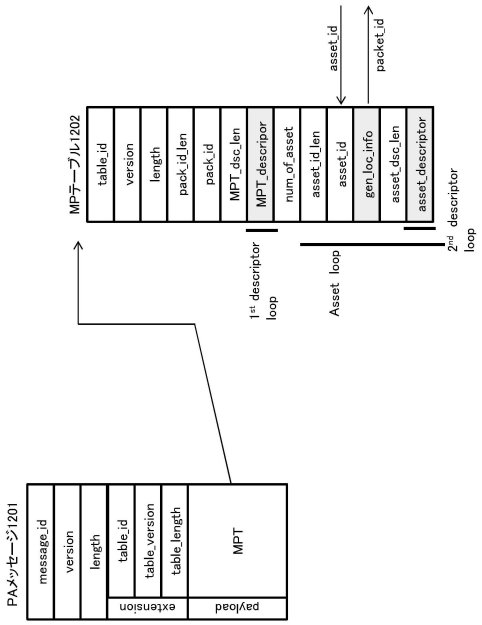
【図１０】


ノンタイムド・メディアを配置したMFUのDU_Header1000
item_ID

【図１１】



【図１２】



【図１３】

PAメッセージ1300

```
PA_message () {
    message_id                                  16      uimsbf
    version                                     8       uimsbf
    length                                      32      uimsbf
    extension {
        number_of_tables            N1          8       uimsbf
        for (i=0; i<N1; i++) {
            table_id                            8       uimsbf
            table_version                       8       uimsbf
            table_length                        16      uimsbf
        }
    }
    message_payload {
        for (i=0; i<N1; i++) {
            table()
        }
    }
}
```

【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１５】

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１６】

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 {			uimsbf
location_count	N6	8	
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１７】

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。この数分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１８】

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
}		

uimsbf

uimsbf

【図１９】

MH AIテーブル(MH AIT)1900

Syntax	No. of bits	Mnemonic
MH-Applicatin_Information_Table(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
application type	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
common_descriptor_length	8	uimsbf
for(i=0; i<N; i++){		
descriptor ()		rpcof
}		
reserved_future_use	4	uimsbf
application_loop_length		
for(i=0; i<N; i++){		
application_identifier ()		
application_control_code	8	uimsbf
reserved_future_use	4	bslbf
application_descriptor_loop_length	12	uimsbf
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	uimsbf
descriptor_length	8	uimsbf
application_profile_length	8	uimsbf
for(i=0;i<N;i++){		
application_profile	16	uimsbf
version_major	8	uimsbf
version_minor	8	uimsbf
version_micro	8	uimsbf
}		
service_bound_flag	1	bslbf
visibility	2	bslbf
reserved_future_use	5	bslbf
application_priority	8	uimsbf
for(i=0;i<N;i++){		
transport_protocol_label	8	uimsbf
}		
}		

【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伝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	uimsbf
descriptor_length	8	uimsbf
protocol_id	16	uimsbf
transport_protocol_label	8	uimsbf
for(i=0;i<N;i++){		
selector_byte	8	uimsbf
}		
}		

【図２３】

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２４】

データ・トランスミッション・・メッセージ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()		
}		
}		

【図２５】

データ・アセット・マネジメント・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
node_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 ()		
}		
}		
descriptor_loop_length	16	uimsbf
for(i=0;i<descriptor_loop_length;i++){		
descriptor()		
}		
}		
}		

【図２６】

データ・ディレクトリーン・マネジメント・テーブル(DDMT)2600

Syntax	No. of bit	Mnemonic
Data_Directory_Management_Table(){		
table_id	8	uimsbf
version	8	uimsbf
length	16	uimsbf
base_folder_path_length	8	uimsbf
for(i=0;i<base_folder_path_length;i++){		
base_folder_path_byte	8	uimsbf
}		
number_of_folder_nodes	8	uimsbf
for(i=0;i<number_of_folder_nodes;i++){		
node_tag	16	uimsbf
folder_node_path_length	16	uimsbf
for(j=0;j<folder_node_path_length;j++){		
forder_node_path_byte	8	char
}		
num_of_files	16	uimsbf
for(j=0;j<num_of_files;j++){		
node_tag	16	uimsbf
file_name_length	8	uimsbf
for (k=0;k<file_name_length;k++){		
file_name_byte	8	char
}		
}		
}		

【図２７】

データ・コンテント・マネジメント・テーブル(DCMT)2700

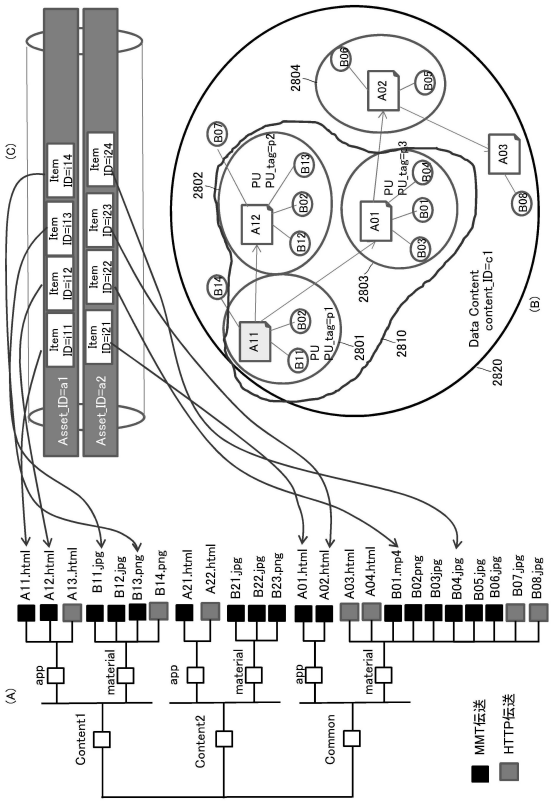
Syntax	No. of bit	Mnemonic
Data_Content_Management_Table(){		
table_id	8	uimsbf
version	8	uimsbf
length	16	uimsbf
number_of_content	8	uimsbf
for(i=0;i<number_of_items;i++){		
contentID	16	uimsbf
content_version	8	uimsbf
content_cache_size	32	uimsbf
number_of_PU	8	uimsbf
for(j=0;j<number_of_PU;j++){		
PU_tag	8	uimsbf
PU_cache_size	32	uimsbf
PU_primary_item_node_tag	16	uimsbf
number_of_PU_member_nodes	16	uimsbf
for(k=0;k<number_of_nodes;j++){		
PU_member_node_tag	16	uimsbf
}		
number_of_pre_cache_nodes	8	uimsbf
for(k=0;k<number_of_pre_cache_nodes;k++){		
pre_cache_node_tag	16	uimsbf
}		
number_of_linked_PU	8	uimsbf
for(k=0;k<number_of_linkedPU;k++){		
linked_PU_tag	8	uimsbf
}		
}		
}		
}		

【図２８】

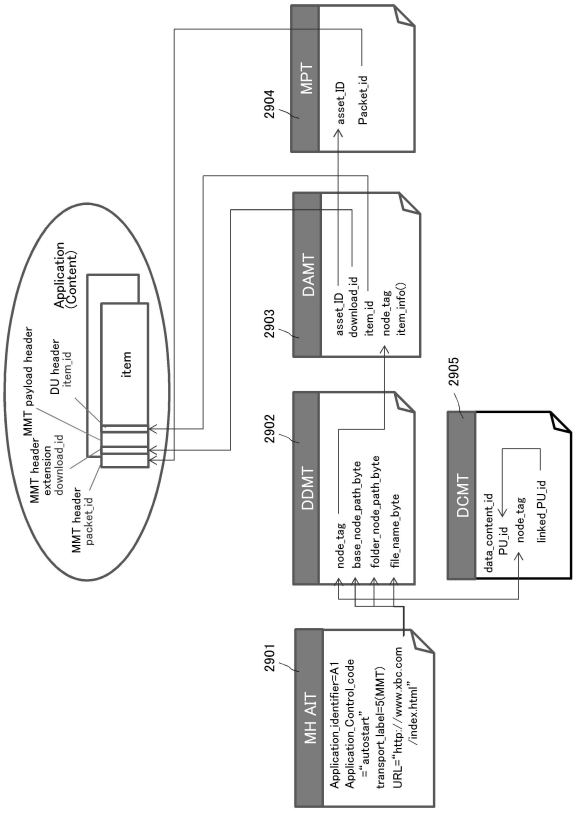
データ・コンテント・マネジメント・テーブル(DCMT)2800

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(i=0;i<number_of_items;i++){		
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
PU_primary_item_node_tag	16	uimsbf
number_of_PU_member_nodes	16	uimsbf
for(k=0;k<number_of_nodes;j++){		
PU_member_node_tag	16	uimsbf
}		
number_of_lock_cache_nodes	8	uimsbf
for(k=0;k<number_of_lock_cache_nodes;k++){		
lock_cache_node_tag	16	uimsbf
}		
number_of_unlock_cache_nodes	8	uimsbf
for(k=0;k<number_of_unlock_cache_nodes;k++){		
unlock_cache_node_tag	16	uimsbf
}		
number_of_linked_PU	8	uimsbf
for(k=0;k<number_of_linkedPU;k++){		
linked_PU_tag	8	uimsbf
}		
}		
}		
}		

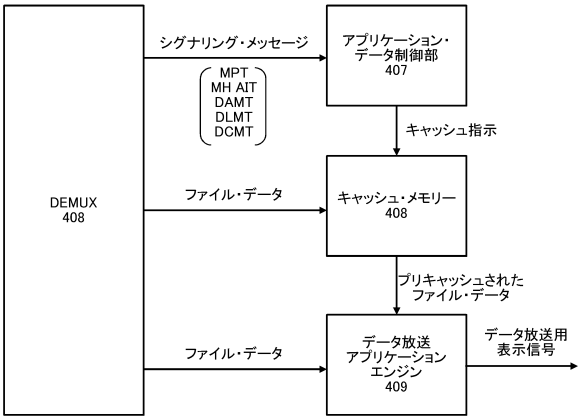
【図２９】



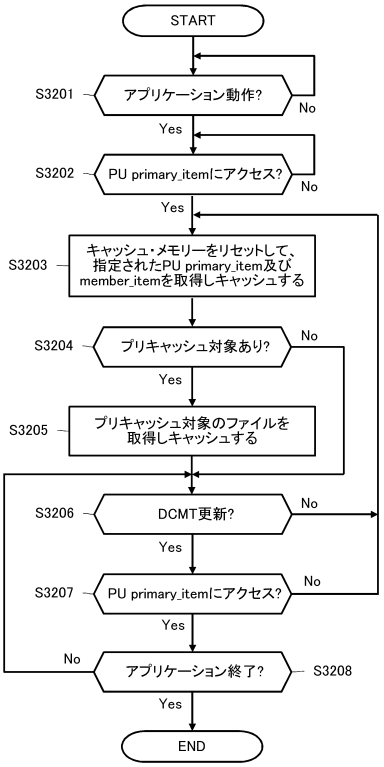
【図３０】



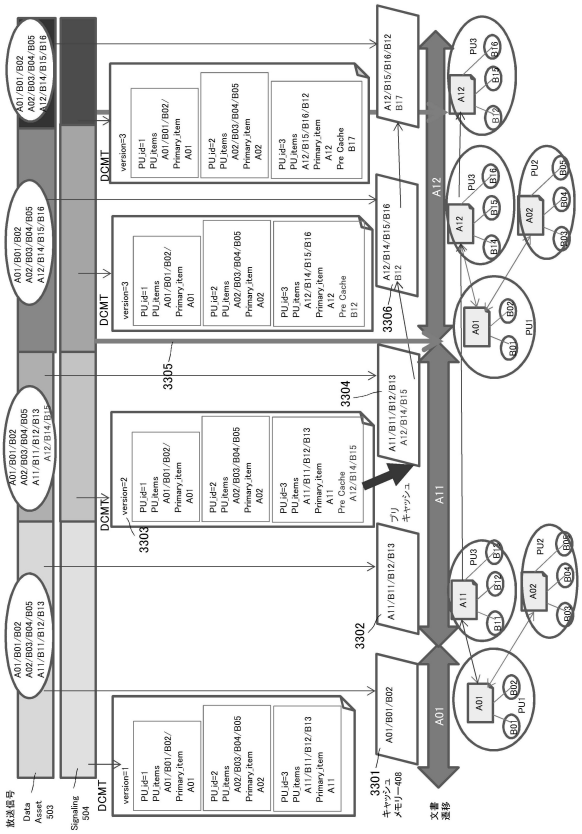
【図３１】



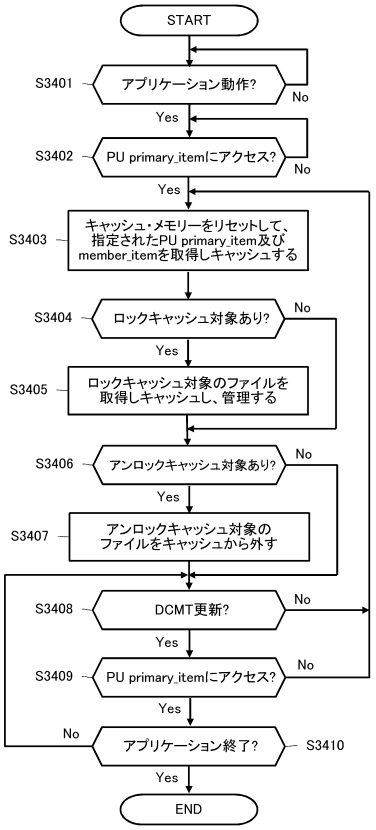
【図３２】



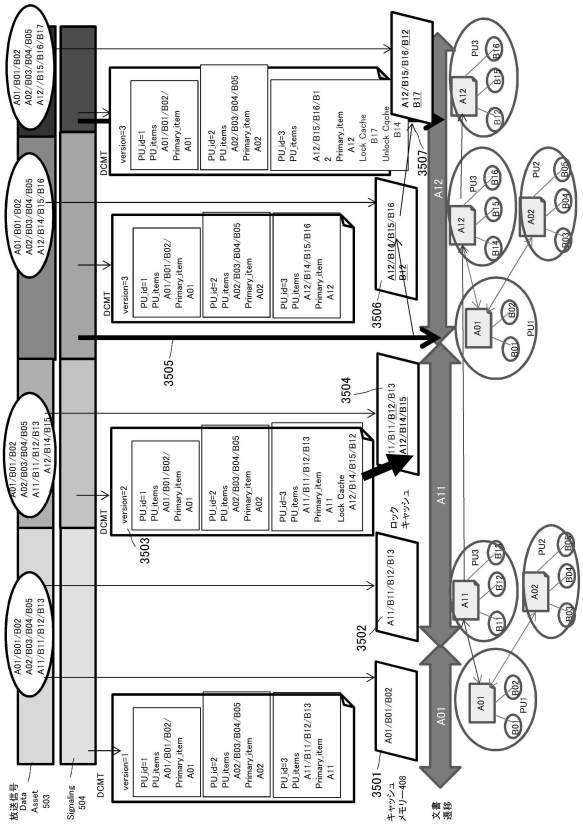
【図３３】



【図３４】



【図３５】


DCMT
version=3
version=2
version=1
PU_id=1
PU_items
PU_id=2
PU_items
PU_id=3
PU_items
Primary_item
Lock Cache
Unlock Cache
3501
3502
3503
3504
3505
3506
3507
A01
A02
A11
A12
PU1
PU2
PU3
B01
B02
B04
B05
B11
B12
B13
B14
B15
B16
B17
拡張境界
Data
Asset
503
Signaling
SVA
キャッシュ
メモリー408
文書
遷移

フロントページの続き

(51)国際特許分類			FI	
H04H	*40/18*	*(2008.01)*	H04H	40/18
H04H	*60/13*	*(2008.01)*	H04H	60/13

グループ株式会社内

(72)発明者　出葉 義治

東京都港区港南１丁目７番１号 ソニーグループ株式会社内

審査官　川中 龍太

(56)参考文献　特開２０１５－１８００５９（ＪＰ，Ａ）

特開２０１０－１３０６３７（ＪＰ，Ａ）

特表２０１３－５１６７０４（ＪＰ，Ａ）

国際公開第２００８／０５３５６８（WO，Ａ１）

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02月28日，Ｖｏｌ．３８ Ｎｏ．１４，ｐ．２９－３２

(58)調査した分野　(Int.Cl.，ＤＢ名)

Ｈ０４Ｎ　２１／００ － ２１／８５８

Ｈ０４Ｈ　２０／００ － ２０／９５

Ｈ０４Ｈ　４０／００ － ４０／９０

Ｈ０４Ｈ　６０／００ － ６０／９８