



## (12) 发明专利

(10) 授权公告号 CN 101366287 B

(45) 授权公告日 2011.09.21

(21) 申请号 200780001943.4

(51) Int. Cl.

(22) 申请日 2007.01.04

H04N 7/66 (2006.01)

(30) 优先权数据

60/743,095 2006.01.05 US

(56) 对比文件

GB 2406483 A, 2005.03.30,

GB 2415873 A, 2006.01.04,

GB 2406488 A, 2005.03.30,

US 2005216472 A1, 2005.09.29,

CN 1468002 A, 2004.01.14,

(85) PCT申请进入国家阶段日

2008.07.04

(86) PCT申请的申请数据

PCT/SE2007/000005 2007.01.04

审查员 张春

(87) PCT申请的公布数据

W02007/078253 EN 2007.07.12

(73) 专利权人 艾利森电话股份有限公司

地址 瑞典斯德哥尔摩

(72) 发明人 T·洛马 M·韦斯特伦德

P·弗罗德

(74) 专利代理机构 中国专利代理(香港)有限公司  
72001

代理人 王岳 陈景峻

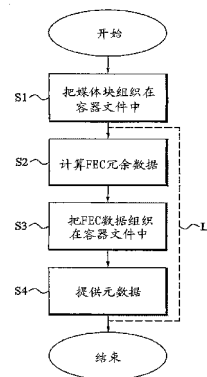
权利要求书 5 页 说明书 27 页 附图 10 页

### (54) 发明名称

媒体容器文件管理

### (57) 摘要

本发明教导一种媒体容器文件(1),其包括被组织到媒体源块(20;22;24)中的媒体数据。对于所述不同的源块(20;22;24)预先计算前向纠错(FEC)冗余数据,并且将其组织到所述容器文件(1)中以作为不同的FEC存储库(30;32;34)。该容器文件(1)还包括元数据(40;45),其提供所述媒体源块(20;22;24)与对应的FEC存储库(30;32;34)之间的关联。该容器文件(1)可以由媒体服务器(200)在媒体会话中采用来在无需大量数据处理和FEC计算的情况下编辑将被传送到发出请求的客户端(400;410;420)的媒体数据分组。



1. 一种生成媒体容器文件的方法,所述方法包括以下步骤:
  - 把至少一个媒体源块组织在所述媒体容器文件中;
  - 基于所述至少一个媒体源块计算前向纠错 FEC 冗余数据;
  - 把所述 FEC 冗余数据组织在所述媒体容器文件内的至少一个 FEC 存储库中;
  - 在所述媒体容器文件中提供元数据,所述元数据提供所述至少一个媒体源块与所述至少一个 FEC 存储库之间的关联;
  - 生成编辑指令,所述编辑指令基于所述元数据定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成数据分组的媒体流;以及
  - 把所述编辑指令组织在所述媒体容器文件中。
2. 根据权利要求 1 所述的方法,还包括以下步骤:
  - 提供媒体源文件;以及
  - 把所述媒体源文件划分成所述至少一个媒体源块。
3. 根据权利要求 2 所述的方法,其中,所述划分步骤包括:至少部分地基于被用来计算所述 FEC 冗余数据的 FEC 算法把所述媒体源文件划分成所述至少一个媒体源块。
4. 根据权利要求 1-3 中任一权利要求所述的方法,其中,所述计算步骤包括以下步骤:
  - 基于将被用来计算所述 FEC 冗余数据的 FEC 算法把所述至少一个媒体源块分割成多个媒体符号;
  - 生成所述媒体源块分割的分割信息;以及
  - 基于所述至少一个媒体源块和所述分割信息计算所述 FEC 冗余数据。
5. 根据权利要求 4 所述的方法,还包括以下步骤:
  - 在所述媒体容器文件中提供所述 FEC 算法的信息和所述分割信息。
6. 根据权利要求 1-3 中任一权利要求所述的方法,还包括以下步骤:
  - 在所述媒体容器文件中提供属性表,该属性表包括所述至少一个媒体源块在所述媒体容器文件内的存储位置信息。
7. 根据权利要求 1-3 中任一权利要求所述的方法,其中,所述生成编辑指令的步骤包括以下步骤:
  - 生成第一编辑指令集,所述第一编辑指令集基于所述元数据定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成具有第一 FEC 冗余开销水平的数据分组的第一媒体流;以及
  - 生成第二编辑指令集,所述第二编辑指令集基于所述元数据定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成具有第二 FEC 冗余开销水平的数据分组的第二媒体流。
8. 根据权利要求 1-3 中任一权利要求所述的方法,其中,所述 FEC 冗余数据是将在媒体会话期间被使用的 FEC 冗余数据,并且所述方法还包括以下步骤:
  - 基于所述至少一个媒体源块计算在后会话修复程序中有用的后会话 FEC 冗余数据;
  - 把所述后会话 FEC 冗余数据组织在所述媒体容器文件内的至少一个 FEC 存储库中;以及
  - 在所述媒体容器文件中提供元数据,所述元数据允许基于与所述至少一个媒体源块

相关联的标识符来识别所述至少一个 FEC 存储库。

9. 一种媒体内容服务器,其包括:

- 媒体块管理器,其被设置成把至少一个媒体源块组织在媒体容器文件中;
- 前向纠错 FEC 编解码器,其被设置成基于所述至少一个媒体源块计算 FEC 冗余数据;
- FEC 数据管理器,其被设置成连接到所述 FEC 编解码器,以便把所述 FEC 冗余数据组织在所述媒体容器文件内的至少一个 FEC 存储库中;
- 元数据管理器,其被设置成在所述媒体容器文件中提供元数据,所述元数据提供所述至少一个媒体源块与所述至少一个 FEC 存储库之间的关联;以及
- 指令管理器,其被设置成:i) 生成编辑指令,所述编辑指令基于由所述元数据管理器提供的所述元数据定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成数据分组的媒体流;以及 ii) 把所述编辑指令组织在所述媒体容器文件中。

10. 根据权利要求 9 所述的媒体内容服务器,还包括:

- 媒体源,其被设置成提供媒体源文件;以及
- 文件划分器,其被设置成连接到所述媒体源,以便把所述媒体源文件划分成所述至少一个媒体源块。

11. 根据权利要求 10 所述的媒体内容服务器,其中,所述文件划分器被设置成至少部分地基于被所述 FEC 编解码器用来计算所述 FEC 冗余数据的 FEC 算法把所述媒体源文件划分成所述至少一个媒体源块。

12. 根据权利要求 9-11 中任一权利要求所述的媒体内容服务器,还包括:

- 块分割器,其用于根据将被所述 FEC 编解码器使用的 FEC 算法把所述至少一个媒体源块分割成多个媒体符号并且用于生成所述媒体源块分割的分割信息,其中,所述 FEC 编解码器被设置成基于所述至少一个媒体源块和所述分割信息计算 FEC 冗余数据。

13. 根据权利要求 12 所述的媒体内容服务器,还包括:

- 信息管理器,其被设置成在所述媒体容器文件中提供由所述 FEC 编解码器使用的所述 FEC 算法的信息和所述分割信息。

14. 根据权利要求 9-11 中任一权利要求所述的媒体内容服务器,还包括:

- 信息管理器,其被设置成在所述媒体容器文件中提供属性表,该属性表包括所述至少一个媒体源块在所述媒体容器文件内的存储位置信息。

15. 根据权利要求 9-11 中任一权利要求所述的媒体内容服务器,其中,所述指令管理器被设置成:i) 生成第一编辑指令集,所述第一编辑指令集基于由所述元数据管理器提供的所述元数据定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成具有第一 FEC 冗余开销水平的数据分组的第一媒体流;以及 ii) 生成第二编辑指令集,所述第二编辑指令集基于由所述元数据管理器提供的所述元数据定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成具有第二 FEC 冗余开销水平的数据分组的第二媒体流。

16. 根据权利要求 9-11 中任一权利要求所述的媒体内容服务器,其中,由 FEC 编解码器生成的所述 FEC 冗余数据是将在媒体会话期间被使用的 FEC 冗余数据,所述 FEC 编解码器被设置成基于所述至少一个媒体源块计算在后会话修复程序中有用的后会话 FEC 冗余数

据,所述 FEC 数据管理器被设置成把由所述 FEC 编解码器计算的所述后会话 FEC 冗余数据组织在所述媒体容器文件内的至少一个 FEC 存储库中,并且所述元数据管理器被设置成在所述媒体容器文件中提供元数据,所述元数据允许基于与所述至少一个媒体源块相关联的标识符来识别所述至少一个 FEC 存储库。

17. 一种媒体会话管理方法,其包括以下步骤:

- 提供媒体容器文件,所述媒体容器文件包括:至少一个媒体源块;至少一个前向纠错 FEC 存储库,其包括基于所述至少一个媒体源块生成的 FEC 冗余数据;元数据,其提供所述至少一个媒体源块与所述 FEC 存储库之间的关联;以及编辑指令,其定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成数据分组的媒体流;

- 通过基于所述元数据和所述编辑指令提取来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据来编辑数据分组;以及

- 在媒体会话期间把所述数据分组传送到至少一个用户终端。

18. 根据权利要求 17 所述的方法,其中,所述编辑指令包括多个编辑指令集,其中每一个编辑指令集与已定义的 FEC 冗余开销相关联,所述方法包括以下步骤:

- 对于所述媒体会话估计 FEC 冗余开销容量;以及

- 基于所述估计的 FEC 冗余开销容量从所述多个编辑指令集中选择编辑指令集,其中,所述编辑步骤包括:通过基于所述元数据和所述选择的编辑指令集提取来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据来编辑所述数据分组。

19. 根据权利要求 17 或 18 所述的方法,其中,所述媒体容器文件还包括属性表,所述属性表包括所述至少一个媒体源块在所述媒体容器文件内的存储信息,并且所述编辑步骤包括:通过基于所述元数据和所述属性表提取来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据来编辑所述数据分组。

20. 根据权利要求 17-18 中任一权利要求所述的方法,还包括以下步骤:

- 接收源自所述至少一个用户终端当中的某一用户终端的针对后会话修复程序请求;

- 基于所述请求从所述至少一个 FEC 存储库中提取 FEC 冗余数据;以及

- 把所述提取的 FEC 冗余数据传送到所述用户终端。

21. 根据权利要求 20 所述的方法,其中,所述媒体容器文件还包括在后会话修复会话期间用于识别 FEC 冗余数据的后会话修复指令,并且所述提取步骤包括:基于所述请求和所述后会话修复指令从所述至少一个 FEC 存储库中提取 FEC 冗余数据。

22. 一种媒体会话服务器,其包括:

- 媒体文件提供器,其用于提供媒体容器文件,所述媒体容器文件包括:至少一个媒体源块;至少一个前向纠错 FEC 存储库,其包括基于所述至少一个媒体源块生成的 FEC 冗余数据;元数据,其提供所述至少一个媒体源块与所述 FEC 存储库之间的关联;以及编辑指令,其定义对来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成数据分组的媒体流;

- 数据分组编辑器,其被设置成连接到所述媒体文件提供器,以便通过基于所述元数据

和所述编辑指令提取来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据来编辑数据分组 ; 以及

- 传送器, 其被设置成连接到所述数据分组编辑器, 以便在媒体会话期间把由所述数据分组编辑器编辑的所述数据分组传送到至少一个用户终端。

23. 根据权利要求 22 所述的媒体服务器, 其中, 所述编辑指令包括多个编辑指令集, 其中每一个编辑指令集与已定义的 FEC 冗余开销相关联, 所述媒体会话服务器还包括 :

-FEC 容量估计器, 其被设置成对于所述媒体会话估计 FEC 冗余开销容量 ; 以及

- 集合选择器, 其被设置成连接到所述 FEC 容量估计器, 以便基于由所述 FEC 容量估计器估计的所述 FEC 冗余开销容量从所述多个编辑指令集中选择编辑指令集, 其中, 所述数据分组编辑器被设置成通过基于所述元数据和由所述集合选择器选择的所述编辑指令集提取来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据来编辑所述数据分组。

24. 根据权利要求 22 或 23 所述的媒体服务器, 其中, 所述媒体容器文件还包括属性表, 所述属性表包括所述至少一个媒体源块在所述媒体容器文件内的存储信息, 并且所述数据分组编辑器被设置成通过基于所述元数据和所述属性表提取来自所述至少一个媒体源块的媒体数据和来自所述至少一个 FEC 存储库的 FEC 冗余数据来编辑所述数据分组。

25. 根据权利要求 22-23 中任一权利要求所述的媒体服务器, 还包括 :

- 接收器, 其被设置成接收源自所述至少一个用户终端当中的某一用户终端的针对后会话修复程序的请求 ; 和

-FEC 管理器, 其被设置成基于由所述接收器接收到的所述请求从所述至少一个 FEC 存储库中提取 FEC 冗余数据,

其中, 所述传送器被设置成连接到所述 FEC 管理器, 以便把由所述 FEC 管理器提取的所述 FEC 冗余数据传送到所述用户终端。

26. 根据权利要求 25 所述的媒体服务器, 其中, 所述媒体容器文件还包括在后会话修复会话期间用于识别 FEC 冗余数据的后会话修复指令, 并且所述 FEC 管理器被设置成基于所述请求和所述后会话修复指令从所述至少一个 FEC 存储库中提取 FEC 冗余数据。

27. 一种后会话修复方法, 其包括以下步骤 :

- 从先前接收了数据分组的用户终端接收针对前向纠错 FEC 冗余数据的请求, 所述数据分组包括从至少一个媒体源块中提取的媒体数据, 所述请求包括与所述至少一个媒体源块相关联的标识符 ;

- 基于所述请求提供媒体容器文件, 所述媒体容器文件包括 : i) 所述至少一个媒体源块 ; ii) 至少一个 FEC 存储库, 其包括基于所述至少一个媒体源块生成的 FEC 冗余数据 ; iii) 元数据, 其允许基于与所述至少一个媒体源块相关联的标识符来识别所述至少一个 FEC 存储库 ; 以及 iv) 后会话修复指令, 其定义对来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑, 以便形成 FEC 冗余数据的至少一个数据分组 ;

- 基于所述接收的标识符和所述后会话修复指令从所述至少一个 FEC 存储库中提取 FEC 冗余数据 ; 以及

- 把所述提取的 FEC 冗余数据传送到所述用户终端。

28. 一种修复服务器, 其包括 :

- 接收器,其用于从先前接收了数据分组的用户终端接收针对前向纠错 FEC 冗余数据的请求,所述数据分组包括从至少一个媒体源块中提取的媒体数据,所述请求包括与所述至少一个媒体源块相关联的标识符;

- 容器文件提供器,其被设置成连接到所述接收器,以便基于由所述接收器接收到的所述请求提供媒体容器文件,所述媒体容器文件包括:i) 所述至少一个媒体源块;ii) 至少一个 FEC 存储库,其包括基于所述至少一个媒体源块生成的 FEC 冗余数据;iii) 元数据,其允许基于与所述至少一个媒体源块相关联的标识符来识别所述至少一个 FEC 存储库;以及iv) 后会话修复指令,其定义对来自所述至少一个 FEC 存储库的 FEC 冗余数据的编辑,以便形成 FEC 冗余数据的至少一个数据分组;

-FEC 数据提取器,其被设置成连接到所述容器文件提供器,以便基于所述接收到的标识符和所述后会话修复指令从所述至少一个 FEC 存储库中提取 FEC 冗余数据;以及

- 传送器,其被设置成连接到所述 FEC 数据提取器,以便把所述提取出的 FEC 冗余数据传送到所述用户终端。

## 媒体容器文件管理

### 技术领域

[0001] 本发明总体涉及媒体和多媒体管理,并且特别涉及产生及使用包含这种媒体和多媒体内容的媒体容器文件。

### 背景技术

[0002] 在最近的几年内,通过不同的网络向客户端提供媒体和多媒体的情况已经大大增多。现在有许多用户采用因特网从媒体服务器访问及下载例如以视频和音频流或文件形式的媒体。这种媒体提供也已经出现在基于无线电的移动通信网络中。当前人们对于多媒体或 TV 内容使用移动网络非常感兴趣。这在本领域中常常被称作移动 TV。所述移动网络中的这种媒体提供当前主要通过单点传送传输来获得。但是,针对移动 TV 的广播/多点传送递送方法当前仍在开发中。这种标准化努力的例子有 3GPP 多媒体广播/多点传送服务 (MBMS) 以及欧洲电信标准协会 (ETSI) 手持式数字视频广播 (DVB-H)。

[0003] 为了迎合针对在不同的有线和无线通信网络中提供媒体的上述日益增长的需求,正在开发可以在无线网络中用于向发出请求的客户端提供媒体内容的流传输和下载服务器。通常的趋势是透明且灵活的流传输/下载服务器,这意味着所述服务器应当最本地包括用来执行不同的媒体管理功能的多个“标准”模块或程序。随后向这些功能提供输入媒体内容以及关于所述模块/程序应当如何处理所述内容的指令。这与在所述服务器中使用固定的预先定义的媒体处理相比将提供更为灵活的媒体供应。

[0004] 与开发灵活的流传输/下载服务器一起,在关于如何在媒体流中引入纠错的领域内同时也有发展。所述多点传送/广播传输是单向的,并且同时寻址大量接收客户端。传统的单点传送可靠性方案(比如自动重复请求 (ARQ)) 不能被缩放来为多点传送/广播传输的大量接收器提供服务。

[0005] 因此,需要引入一种与多点传送/广播媒体传输相关联的可靠性方案。引入这种可靠性方案还应当符合灵活的流传输和下载服务器解决方案的趋势。

### 发明内容

[0006] 本发明克服了现有技术设置的上述和其他缺陷。

[0007] 本发明的一般目的是提供一种可以被用在多媒体会话中的媒体容器文件。

[0008] 本发明的另一个目的是提供一种还可以被用在后会话 (post-session) 修复程序中的媒体容器文件。

[0009] 如所附权利要求书所限定的那样,本发明满足了上述和其他目的。

[0010] 简而言之,本发明涉及到生成及使用媒体容器文件,并且涉及到用于生成及使用这种容器文件的设备。

[0011] 生成媒体容器文件涉及到提供至少一个媒体源文件,其包括将被传送到发出请求的客户端并且将在所述客户端处被呈现的媒体或多媒体数据。该容器文件被视为包括一个或多个媒体源块,这取决于所述源文件的大小。根据本发明处理至少一个所述媒体源块,以

便根据该源块计算 FEC 冗余数据或符号。因此,该源块的媒体数据被输入到 FEC 算法,以便计算至少一个 FEC 符号。优选地对于源文件的每一个媒体源块执行该 FEC 符号计算。随后把该至少一个媒体源块组织到所述媒体容器文件中。相应地,也把所计算的 FEC 数据组织到所述媒体容器文件内的一个或多个 FEC 存储库中。每一个所述 FEC 存储库包括为特定媒体源块计算的 FEC 数据。提供元数据并且将其包括在所述容器文件中,以便提供媒体源块与其 FEC 存储库之间的关联。

[0012] 所得到的容器文件可以由媒体服务器在媒体会话期间采用,以便利用该容器文件中的元数据来编辑包括媒体数据和 FEC 数据的数据分组。预先计算 FEC 数据并且把媒体和 FEC 数据组织在本发明的容器文件中,允许所述媒体服务器以简单的在计算方面并不昂贵的方式把媒体数据和 FEC 数据插入到将被传送到发出请求的客户端的数据分组中,而无需进行大量数据处理以及计算需求高的 FEC 计算。

[0013] 在一个优选实施例中,所述容器文件还包括编辑指令,所述媒体服务器在编辑包含来自所述容器文件的媒体和 FEC 数据的数据分组时使用并且遵循所述编辑指令。在这种情况下,所述容器文件包括能够按照可靠的方式成功地把所述媒体数据转送到所述客户端所需要的所有媒体内容、保护数据和指令。

[0014] 本发明的容器文件还可以被用在后会话修复程序中,其中尽管把 FEC 冗余保护数据包括到所传送的数据分组中,但是客户端仍然不能成功地在媒体会话期间接收所有媒体数据。在这种情况下,由修复服务器使用在前一个媒体会话期间所使用的相同容器文件的副本。该服务器能够根据来自所述客户端的请求从所述容器文件内的其中一个 FEC 存储库检索 FEC 冗余数据。随后把该 FEC 数据返回到该客户端,从而允许该客户端成功地呈现所有媒体数据。

## 附图说明

[0015] 可以参照结合附图进行的以下详细描述来更好地理解本发明及其其他目的和优点,其中:

[0016] 图 1 是示出了根据本发明的一方面的生成媒体容器文件的方法的流程图;

[0017] 图 2 是示出了图 1 的文件生成方法的附加步骤的流程图;

[0018] 图 3 是示出了图 1 的文件生成方法的附加步骤的流程图;

[0019] 图 4 是示出了图 1 的文件生成方法的附加步骤的流程图;

[0020] 图 5 是示出了根据本发明的另一方面的媒体会话管理方法的流程图;

[0021] 图 6 是示出了图 5 的会话管理方法的附加步骤的流程图;

[0022] 图 7 是更加详细地示出了图 5 的编辑和传送步骤的一个实施例的流程图;

[0023] 图 8 是更加详细地示出了图 5 的编辑和传送步骤的另一个实施例的流程图;

[0024] 图 9 是示出了图 5 的会话管理方法的附加步骤的流程图;

[0025] 图 10 是示出了根据本发明的另一方面的后会话修复方法的流程图;

[0026] 图 11 是根据本发明的另一方面的媒体容器文件的示意性总览;

[0027] 图 12 是示出了对采用根据本发明的媒体容器文件的不同媒体流进行编辑的示意图;

[0028] 图 13 是示出了对采用根据本发明的媒体容器文件的不同媒体流进行编辑的另一

个示意图；

[0029] 图 14 是根据本发明的包括管理媒体容器文件的服务器的通信网络的总览；

[0030] 图 15 是根据本发明的另一方面的内容服务器的示意性方框图；

[0031] 图 16 是更加详细地示出了图 15 的容器文件产生器的一个实施例的示意性方框图；

[0032] 图 17 是根据本发明的另一方面的媒体会话服务器的示意性方框图；以及

[0033] 图 18 是根据本发明的另一方面的修复服务器的示意性方框图。

## 具体实施方式

[0034] 在附图中，相同的附图标记将被用于相应的或类似的元件。

[0035] 本发明总体上涉及媒体和多媒体数据管理，并且特别涉及产生及利用与基于无线电的通信网络中的媒体服务器（比如流传输或下载服务器）相关联的容器文件。除了将要传送到发出请求的一个（或多个）客户端的媒体内容以及用于在所述媒体服务器中执行媒体处理和传输的指令之外，本发明的这些媒体容器文件还包括为所述媒体会话提供可靠性保护的数据。由于存在被包括在所述容器文件中的预先生成的前向纠错（FEC）冗余数据，因此该可靠性保护是可以实现的。

[0036] 如本领域中已知的那样，FEC 涉及到把冗余数据添加到所传送的有效载荷数据中，这样允许接收器在无需向发送方请求附加数据的情况下检测并校正错误。FEC 的优点在于常常能够避免对数据的重传，但是其代价则是平均而言更高的带宽要求。因此，FEC 可以被有利地用于媒体内容的基于多点传送 / 广播的递送，其中重传将很难实现。

[0037] FEC 是通过利用预定算法或方案向将被传送的信息添加冗余度而实现的，所述算法或方案在本领域中通常被称作 FEC 编解码器。每一个这种冗余比特都是许多原始信息或有效载荷比特的复杂函数。在输出中包括未经修改的输入的 FEC 编解码器被称作系统编解码器。换句话说，(N,K) 系统 FEC 编解码器保留 K 个源符号或有效载荷符号并且附加 (N-K) 个 FEC 符号。相应地，(N,K) 非系统 FEC 编解码器从 K 个源符号产生 N 个 (FEC 或源) 符号，而不必保留所有 K 个源符号。

[0038] 主要有两类 FEC 编解码器：块编解码器和卷积编解码器。FEC 块编解码器在由预定尺寸的比特或符号构成的固定尺寸的块（分组）上进行操作，而卷积编解码器则在任意长度的比特或符号流上进行操作。DigitalFountains Raptor 编解码器是能够从单一源块中产生任意数目的 FEC 冗余符号的 FEC 块编解码器。这是该 FEC 编解码器的一个有利属性，这是因为在所述源块构造中没有任何改变的情况下可以生成不同的保护 - 开销配置。Reed-Solomon 是另一种 FEC 块编解码器，但是其需要对于不同的保护开销尺寸在源块分割中做出改变。FEC 块编解码器的其他例子包括 Golay、BCH (Bose、Ray-Chaudhuri、Hocquenghem) 和 Hamming。与本发明相结合地使用的一种优选的 FEC 编解码器是 Digital Raptor 编解码器。

[0039] 根据本发明，媒体或多媒体数据或内容指代可以由内容提供商或服务器提供给客户端以供呈现数据的任何数据。典型的优选实例包括视频数据和音频数据。所述数据可以具有预先编码的固定速率音频或视频内容版本的形式，或者具有可扩展音频或视频内容的形式。其他媒体实例包括静止图像 (JPEG)、位图图形 (GIF 和 PNG)、矢量图形 (SVG) 以及合

成音频 (SP-MIDI) 和文本 (XHTML 和 SMIL)。

[0040] 图 1 是根据本发明的生成媒体容器文件的方法的流程图。该媒体容器文件可以被视为完整的输入包,其由媒体服务器在媒体会话期间使用来提供媒体内容并且把媒体数据形成到可传送的数据分组中。因此,除了媒体内容本身之外,所述容器文件优选地包括为了由所述媒体服务器在媒体会话期间执行所述处理并且允许传送所述媒体内容所需要的信息和指令。

[0041] 所述方法开始于步骤 S1,在该步骤中,至少一个媒体源块被组织并且存储在所述容器文件中。如果在该容器文件中存在多于两个媒体源块,则把它们视为相同媒体内容文件或流(例如视频流)以及/或者不同的媒体文件或流(例如视频流的几个媒体源块以及相应的相关联的音频流的几个媒体块)的单独媒体块。所述至少一个媒体源块包括应当被发送到客户端的媒体数据或符号,其在该客户端中被呈现以便把媒体内容呈现给用户。所述媒体块可以具有固定的相同尺寸,或者至少其一部分,如果多于一个的话,可以具有不同的比特/符号尺寸。

[0042] 在步骤 S1 中被组织在所述容器文件中的所述至少一个媒体块优选地共同包括将在媒体会话期间被传送到客户端的所有媒体内容数据。换句话说,所述容器文件包含用于整个多媒体呈现的媒体数据。因此,如果所述媒体内容包括音乐视频,则所述容器文件优选地包括具有视频数据的媒体源块和具有相应的音频数据的媒体块。但是本发明预期可以在多个可能的媒体版本中提供同一项媒体内容。例如,所述音乐视频的视频部分可以被提供在多个预先编码的视频版本中,其中每一个所述视频版本被适配成与给定的带宽或者比特率水平或区间相结合地使用。因此在所述容器文件中可以有给定媒体内容的多个版本。在这种情况下,每一个所述媒体版本可以被视为包括一个或多个媒体源块。虽然在所述容器文件中可以有多个媒体版本,但是在所述媒体会话期间的给定时刻通常仅仅使用一个所述版本,但是例如基于可用带宽水平的改变,在所述会话期间,在各媒体版本之间可以有切换。

[0043] 为了给所述容器文件中的所述媒体内容提供可靠性保护,下一个步骤 S2 基于在步骤 S1 中被组织在所述容器文件中的各媒体块当中的至少一个媒体源块预先计算 FEC 冗余数据。在该计算步骤 S2 中,优选地采用在媒体块的基础上进行操作的 FEC 块编解码器(比如 digital fountainraptor 编解码器)。但是也可以采用卷积 FEC 编解码器,所述卷积 FEC 编解码器也在本发明的范围内。在一种优选实现方式中,对于所述至少一个媒体源块生成一组 FEC 冗余符号。该 FEC 符号组可以包括一个(但是优选地包括多个)基于所述媒体源块的源符号计算的 FEC 符号。将为某一媒体源块计算的 FEC 符号的数目可以由所采用的 FEC 编解码器中的限制来限定,其可以是所述媒体源块中的媒体源符号的数目的函数或者由某种其他标准来限制(例如所述容器文件的尺寸限制)。这些预先计算的 FEC 冗余符号的一般思想是在所述容器文件中提供可以获得的并且可以在媒体会话期间由媒体服务器使用来在该媒体会话中提供可靠性保护的 FEC 符号的存储库。因此,每一个媒体源块的 FEC 符号的数目优选地是根据这一标准确定的;即能够提供可靠性保护。

[0044] 除了在媒体会话期间可用于提供可靠性保护之外,在步骤 S2 中计算的 FEC 冗余符号还可以被用在后会话修复程序或会话中。可替换地,所述 FEC 冗余符号的专用子集在所述媒体会话期间可用于可靠性保护,并且所述冗余符号的另一个子集被专用于后会话修

复,在本文中将更加详细地描述这一点。因此,本发明的优点在于,相同的容器文件(或其拷贝)既可以由媒体服务器在媒体会话期间使用又可以由修复服务器在修复会话期间使用。这样允许与媒体管理相关联的高度灵活性。关于 FEC 编解码器的更多信息可以参照文献 [1] 的附录 B,该附录 B 的教导在此被引入以作参考。

[0045] 下一个步骤 S3 把在步骤 S2 中计算的所述 FEC 冗余数据与所述媒体源块一起组织在所述容器文件内。该步骤 S3 优选地把所述媒体源块的 FEC 冗余符号存储为所述容器文件内的存储库。如果已经出于可靠性保护和后修复的目的生成了 FEC 冗余符号,则这些符号可以被提供在相同的 FEC 存储库中。可替换地,第一存储库容纳用于可靠性目的的 FEC 符号,第二存储库包含后修复 FEC 符号。

[0046] 下一个步骤 S4 提供将被包括在所述容器文件中的元数据。该元数据提供在步骤 S1 中被添加到所述容器文件中的媒体源块与在步骤 S3 中被存储在所述文件中的 FEC 冗余数据之间的关联。这种关联可以具有从所述媒体源块在所述文件内的存储位置到所述 FEC 存储库的存储位置的指针的形式,或者反之亦然。因此,该元数据允许在给出特定媒体源块或者其在所述容器文件内的位置的情况下识别出基于该媒体源块计算的相关联的 FEC 冗余数据或者识别出该冗余数据在所述文件内的存储位置。在所述媒体源块和 / 或相关联的 FEC 存储库被存储在所述容器文件中的预先定义的“标准”位置处的情况下,取代采用指针,所述元数据可以包括所述媒体源块和 / 或相关联的 FEC 存储库的标识符。随后使用该元数据来识别出所述文件中的媒体源块和 FEC 存储库的其中之一,并且基于该识别出的位置可以识别出所述媒体源块和 FEC 存储库当中的另一个。

[0047] 如果每一个媒体源块对应至少两个 FEC 存储库(用于会话中和后会话使用),则所述元数据优选地包括提供在所述媒体源块与全部两个 FEC 存储库之间的关联的信息。这例如可以通过针对所述媒体源块把会话中元数据和后会话元数据分别包括在所述容器文件中来实现。

[0048] 在本发明的一种典型的实现方式中,在步骤 S1 中把多个媒体源块组织到所述容器文件中。在这种情况下,优选地对于每一个这种媒体源块或者至少对于多组媒体源块重复步骤 S2 到 S4,这由线 L1 示意性地示出。因此,如果在步骤 S1 中把 N 个媒体源块组织到所述容器文件中,则优选地把步骤 S2 到 S4 重复 N 次,这意味着除了所述源块之外还把至少 N 个 FEC 存储库和 N 个元数据版本组织在所述容器文件中。

[0049] 本发明预期所生成的媒体容器文件可以包含完整媒体会话所需要的所有媒体数据和 FEC 数据。然而,所述媒体数据和 FEC 数据实际上可以被应用在多个不同的会话中。例如,所述容器文件可以包括音乐视频、足球比赛等等的媒体数据。在这种情况下,媒体服务器不必传送被包含在所述 FEC 存储库中的所有媒体数据,而可以仅仅传送由客户端所请求的特定数据。但是另一个媒体服务器可能接收了相同容器文件的副本,并且替换地把该文件中的其他媒体数据提供给它客户端。

[0050] 所述方法随后结束。

[0051] 上面结合图 1 描述的容器文件生成优选地是在可以访问内部或外部媒体内容源的内容产生器或者服务器处实施的。随后可以在诸如计算机存储器的存储介质中或者在诸如电信号或无线电信号的物理信号中表示所生成的容器文件,以便例如在本地系统内传输或者通过本地或全球网络传输。在一个典型的实施例中,所述容器文件被提供为到媒体服

务器的无线电信号,以便使用在与不同客户端的媒体会话中。替换地或附加地,所述容器文件可以被提供到修复服务器,该修复服务器可以由客户端在与所述媒体服务器的媒体会话之后访问,以便在必要时对所接收的媒体内容进行后会话修复。

[0052] 在下面的公开内容中将通篇使用术语“媒体容器文件”,其含义包括用于存储在存储介质中的数据文件以及用于传送或分发的信号。

[0053] 图 2 是示出了图 1 的容器文件生成方法的附加步骤的流程图。所述方法开始于步骤 S10,在该步骤中提供至少一个媒体源文件。在该说明性实施例中,可以在所述容器文件产生器处以包含所述媒体数据的源文件或流的形式获得媒体内容。在该步骤 S10 中,可以提供一个或多个媒体源文件以便包括到容器文件中。例如,第一媒体文件可以包含视频数据,而第二相关联的文件则包含音频数据。为了允许高效地计算 FEC 冗余数据以及随后使用这种 FEC 数据,在步骤 S10 中提供的所述一个(或多个)媒体源文件在下一个步骤 S11 中被划分成多个媒体块。该媒体源块于是可以被视为所述媒体源文件的一个片段,其中可以对该片段应用 FEC 代码。在源符号或者数据比特方面的所述媒体块的尺寸可以结合所述划分被预先定义或选择。在前一种情况下,所述尺寸应当由将被应用来计算 FEC 冗余数据的预定 FEC 方案或代码来定义。因此,实际的 FEC 编解码器或算法以及 / 或者所需要的 FEC 保护开销可能会影响所述媒体文件划分和媒体块尺寸。

[0054] 在所述输入媒体源文件的比特或符号尺寸小于可以由 FEC 编解码器有效处理的最大尺寸时,当然不需要把该源文件划分成媒体源块,并且可以省略步骤 S11。于是将该输入媒体源文件视为根据本发明的媒体源块。

[0055] 应当注意的是,即使存在优选的块尺寸,也并不需要从媒体源文件生成的所有媒体源块都具有该优选尺寸。例如,与其他等同块相比,最后一个媒体源块的尺寸可能较小,这是因为所述媒体文件的剩余部分不包含足够的媒体数据以达到所述优选块尺寸。

[0056] 上述划分步骤 S11 不一定意味着把所述媒体源文件物理地划分成被存储在所述容器文件中的单独位置处的单独媒体源块。与此相对,在大多数实际的实现方式中,所述媒体源文件在所述容器文件中被存储为一个连续数据序列,但是被视为或者被虚拟地划分成媒体源块。例如,可以把包含  $2N$  个源符号的媒体源文件划分成使得源符号  $[0, N-1]$  属于第一源块,并且使得符号  $[N, 2N-1]$  属于第二源块。

[0057] 所述方法随后继续到图 1 的步骤 S1,在该步骤中把所述一个(或多个)媒体源块组织到所述容器文件中。

[0058] 图 3 是示出了图 1 的容器文件生成方法的附加步骤的流程图。该方法开始于步骤 S20,在该步骤中把媒体源块分割成源符号或者所谓的组块(chunk)。这些符号通常包括几百个字节。该块分割优选地是至少部分地基于将被用来为当前媒体源块计算 FEC 符号的 FEC 编解码器 / 算法的信息来执行的。如前所述,基于 Reed-Solomon 的 FEC 编解码器需要基于所期望的保护开销尺寸(即 FEC 冗余符号的数目)而改变所述源块分割。

[0059] 因此,实际的 FEC 编解码器或算法以及 / 或者所需的 FEC 保护开销可能会影响所述媒体源分割以及媒体符号尺寸。此外,在步骤 S20 的该源块分割中可以使用由媒体服务器使用来传送所述媒体内容的其他参数,比如数据分组的尺寸,其中所述数据分组比如是用户数据报协议(UDP)分组。在这种情况下,可以限制所述源符号的尺寸,从而可以把至少一个完整源符号置于 UDP 分组中。

[0060] 该分割步骤 S20 不一定意味着把所述媒体源块物理地划分成被存储在所述容器文件中的单独位置处的单独源符号。与此相对,在大多数实际的实现方式中,所述媒体源文件在所述容器文件中被存储为一个连续数据序列,但是被视为或者被虚拟地划分成媒体源块,所述媒体源块又被虚拟地分割成源符号。

[0061] 在下一个步骤 S21 中生成所述分割的信息。该信息基本上表明所述数据序列的哪些部分属于所述媒体源块的哪一个源符号。所述分割信息可以被组织到一个表中,该表表明媒体源块 Z 的比特 X 到比特 Y 构成源符号。可替换地,所述信息可以包括每一个源符号的以字节计的尺寸。于是在知道媒体源块在媒体文件中的起始位置的情况下有可能确定属于不同源符号的数据部分。

[0062] 该方法随后继续到步骤 S2,在该步骤中把所述分割符号与所述媒体源块一起用于计算所述 FEC 数据。因此,所述分割信息允许识别出应当被输入到所述 FEC 编解码器以便生成 FEC 符号的媒体源块的各数据部分。

[0063] 如前所述,本发明的一个目的是提供一种媒体容器文件,其除了实际的媒体数据之外还包括可靠性保护(即预先计算的 FEC 数据)。这意味着所述文件到源块划分、块分割以及 FEC 保护计算是“离线”进行的,并且与媒体服务器中的实际的媒体传输处理无关。这种预处理简化了所述服务器的任务,并且降低了所述服务器的性能要求和复杂度。此外,所述容器文件优选地还包括由媒体服务器识别出媒体数据和 FEC 数据并且把所述媒体数据和 FEC 数据组成媒体流所需的信息和指令,其中所述媒体流可以被传送到发出请求的客户端。所述容器文件优选地还包括由修复服务器识别并且组成 FEC 数据所需要的信息和指令,其中所述 FEC 数据可用于完成媒体会话之后的后修复处理。因此,所述容器文件可以被视为能够被透明且灵活的服务器用于数据编辑和传输的数据、信息和指令的完整包。

[0064] 图 4 是示出了图 1 的容器文件生成方法的附加步骤的流程图。该方法从图 1 的步骤 S4 继续。在下一个步骤 S30 中,提供被采用来计算 FEC 冗余数据的方案的 FEC 算法的信息。该信息可以具有特定算法名称或者某种其他描述性信息的形式。在一种替换方法中,每一种可用的 FEC 算法都具有预先定义的标识符。因此,可以随后在步骤 S30 中仅仅提供该 FEC 标识符。在一种典型的实现方式中采用了相同的 FEC 编解码器来确定对于媒体源文件的所有媒体源块的冗余数据。但是实际上有可能对于给定媒体源文件的不同媒体源块或者对于不同源文件的媒体源块使用不同的 FEC 编解码器。因此,所述 FEC 信息可以表明利用了单一 FEC 编解码器来处理所述容器文件内的所有块以生成 FEC 符号,或者标识出对于哪一个源块或者源块组使用了不同的 FEC 编解码器。

[0065] 在下一个步骤 S31 中提供特定媒体源文件划分的信息。当所述媒体服务器或修复服务器要向发出请求的客户端提供媒体数据分组流或者 FEC 数据时,该信息对于所述媒体服务器或修复服务器具有相关性。

[0066] 在下一个步骤 S32 中优选地提供属性表。如果在所述容器文件中包括多于一个媒体源文件/流,则该属性表是特别有用的,但是在仅仅包含单一媒体源文件时也可以有利地使用该属性表。该文件属性表通常包含所述媒体源文件的媒体类型的信息,优选地是所述媒体的多目的因特网邮件扩展(MIME)类型。因此,该 MIME 信息可以表明所述媒体是音频媒体、视频媒体或者某种其他媒体类型,其中包括同步多媒体集成语言(SMIL)。该 MIME 类型向所述媒体服务器提供关于什么类型的数据被实际包括在所述容器文件中的信息。所

述属性表还可以包括被用于所述媒体数据的任何编码方案的信息,其中包括 gzip。在所述属性表中还可以包括尺寸信息。该尺寸信息可以声明每一个媒体源文件的在字节或符号数目方面的总尺寸、所述一个(或多个)源文件的媒体源块的对应尺寸(基本上对应于在步骤 S31 中提供的所述划分信息)、对于将在传送数据时被使用的数据分组的最大或目标有效载荷尺寸、媒体源符号(基本上对应于在图 3 的步骤 S21 中生成的所述分割信息)和/或 FEC 符号的(以字节计的)尺寸等等。对于被包括在所述容器文件中的每一个媒体源文件,在所述属性表中优选地包括文件名或文件标识符。

[0067] 优选地可以在所述属性表中找到关于每一个媒体源文件在所述容器文件中的实际存储位置的信息。该位置信息可以表明该源文件的第一媒体源块的起始位置,并且随后在所述容器文件中的该位置之后找到剩余的媒体块。在图 1 的步骤 S4 中生成的提供所述容器文件中的媒体源块与 FEC 存储库之间的关联的元数据也可以被包括在所述属性表中。相应地,关于所采用的块分割以及对于 FEC 冗余数据计算所采用的 FEC 代码的信息优选地被包括在该表中。

[0068] 因此,所述容器文件的属性表可以被用于媒体源的单一信息源,以便定位相关的媒体源文件/块并且提供对于该源块的 FEC 冗余数据以及可用于在媒体会话期间编辑媒体分组的其它信息。

[0069] 下一个步骤 S33 生成供媒体服务器使用的编辑指令。这些指令被用来基于所述元数据定义编辑,所述元数据提供了来自所述媒体源块的媒体数据与来自相关联的一个(或多个)FEC 存储库的 FEC 冗余数据的 FEC-块关联以便形成数据分组的媒体流。因此,这些指令可以被视为提示或元数据,其提供关于如何使用被包括在所述容器文件中的(媒体和 FEC)数据来组成具有可靠性保护的可传送媒体分组流的指令。因此,这些指令被与所述关联元数据一起使用,以便把媒体数据和 FEC 数据一起编辑成适当的分组,以供在媒体会话期间传输到发出请求的客户端。因此,所述指令将描述媒体源数据和 FEC 数据的服务器侧传输顺序。但是应当注意到,所述指令通常不包括时间调度信息、目标/源地址或端口的信息或者其他特定于会话的信息。这意味着所述容器文件以及其中的编辑指令对于特定会话来说是透明的,并且实际上可以被媒体服务器用于与不同接收客户端的多个不同会话,但是也可以被不同媒体服务器使用。

[0070] 所述编辑指令可以应用于所述媒体源块和 FEC 存储库的子集,这意味着必须由媒体服务器在会话期间读取并使用多条这种指令。可替换地,编辑指令包括对于单一媒体源文件或者实际上对于所述容器文件中的所有媒体源文件所需要的所有信息。

[0071] 在步骤 S33 中可以实际生成多于一个的编辑指令集合。在这种情况下,可以提供不同的替换指令,从而媒体服务器可以选择确定对于特定媒体会话采用哪一个特定指令集。例如,在采用单一传输信道以用于数据传输时,第一编辑指令可以被用于描述媒体源块和 FEC 数据的传输顺序。如果有多条信道可用(即意味着可以并行地而不是顺序地传送数据),则第二指令可以被应用于相同的媒体源块和 FEC 数据,但是其提供编辑和传输顺序信息。因此,可以使用几条编辑指令来提供针对不同传输信道条件的替换传输会话。

[0072] 按照类似的方式,对于不同的可靠性保护开销可以包括替换的编辑指令。例如,第一编辑指令被用于针对第一最大保护开销水平描述媒体源块和 FEC 数据的编辑和传输顺序,而第二指令则被用于具有第二不同 FEC 开销水平的相同媒体源块。如果该第二 FEC 开

销水平高于（低于）第一水平，则可以向给定数量的媒体源符号添加更多的 FEC 符号或其在在本领域中也称作的奇偶符号。

[0073] 除了将被媒体服务器使用的编辑指令之外，可以在步骤 S33 中生成适用于修复服务器的编辑指令。这些专用于修复的指令随后主要由后会话修复服务器采用，以便允许识别出 FEC 冗余数据（可能的专用后会话 FEC 冗余数据）以便传送到发出请求的客户端，其中该客户端未能正确地接收及解码所有在前一个媒体会话期间所接收的媒体数据。

[0074] 下一个步骤 S34 在容器文件中组织在先前的步骤 S30 到 S33 以及优选地还有图 3 的步骤 S21 中提供和生成的信息、表和指令。所述容器文件于是将包含由媒体服务器或修复服务器识别及组成用于传输到发出请求的一个（或多个）客户端的数据所需要的“原始”媒体数据、FEC 数据和元数据的完整集合。所述方法随后结束。

[0075] 图 11 是根据本发明的媒体容器文件 1 的示意性总览。如前所述，该容器文件 1 包含多个媒体源文件 10、12、14 的媒体数据，在该图中示出了 M 个这种文件，其中  $M \geq 1$ 。每一个文件 10、12、14 的媒体数据被视为划分成多个媒体源块 20、22、24。在该图中对于第一媒体源文件 10 示出了  $Q_1$  个这种块 20、22、24，其中  $Q_1 \geq 1$ 。每一个这种媒体源块 20、22、24 又被视为分割成源符号。

[0076] 除了具有媒体数据的媒体源块 20、22、24 之外，所述容器文件 1 包括 FEC 存储库 30、32、34，其包含将与所述媒体数据相结合地使用来提供可靠性保护的预先计算的 FEC 冗余数据。在一种优选的实现方式中，每一个媒体源块包括至少一个专用的 FEC 存储库 30、

32、34。在这种情况下，该图中的 FEC 存储库 30、32、34 的数目 N 是  $N \geq \sum_{i=1}^M Q_i$ 。在一种替换方法中，每一个媒体源文件 10、12、14 具有至少一个专用 FEC 存储库，即  $N \geq M$ 。在这种情况下，所述存储库 30、32、34 的对应的区段或区域可以被不同的媒体源块 20、22、24 采用。

[0077] 本发明的关联元数据 40 也被提供在所述容器文件 1 中，所述关联元数据 40 提供所述 FEC 存储库 30、32、34 与所述一个（或多个）媒体源块 20、22、24 之间的关联，其中基于所述关联元数据 40 来计算所述存储库 30、32、34 中的 FEC 数据。图 11 示出了该元数据 40 在文件 1 中的多个不同的可能位置。在第一实施例中，与所述一个（或多个）相关联的媒体源块 20、22、24 相结合地存储所述元数据。因此，通过识别出所述文件中的媒体源块 20、22、24 还允许识别出该源块 20、22、24 的对应元数据 40。替换地或附加地，把所述元数据 40 与对应的 FEC 存储库 30、32、34 一起存储。因此，每一个存储库 30、32、34 具有相连接的关联元数据 40，其允许识别出为之应用特定存储库 30、32、34 的一个（或多个）相关媒体源块 20、22、24。如果所述容器文件 1 包括优选的文件属性表 60，则所述关联元数据 40 可以被提供在该属性表中。在这种情况下，媒体服务器可以仅仅调查该文件属性表 60 以便识别出将在媒体会话期间使用的相关媒体数据和 FEC 数据的位置。在另一种可能的实现方式中，与所述容器文件 1 的不同编辑指令 50、52、54 相结合地存储所述关联元数据 40，所述编辑指令 50、52、54 在该图中被标记为提示轨道（hinttrack）。在这种情况下，每一条提示轨道 50、52、54 仅仅需要包含可以由该提示轨道 50、52、54 中的指令实现的媒体会话所需要的元数据 40。多个此类可能的存储位置的组合也是可能的，并且落在本发明的范围之内。

[0078] 在本发明的一种可选但是优选的实现方式中，所述容器文件 1 还包括专门在修复程序期间使用的编辑指令 70，其在该图中被标记为修复提示轨道 70。该提示轨道 70 于是

优选地包括关联元数据 45, 所述关联元数据 45 可用于识别包括可以用在所述修复程序中的后会话 FEC 数据的 FEC 存储库 30、32、34, 这将在本文中进一步进行描述。

[0079] 根据本发明的一个特定实施例, 所述媒体容器文件 1 是交织的单元, 其针对循序 (progressive) 下载或流传输而被优化。从而可以通过到发出请求的客户端的所谓的循序下载或流传输来传送或下载整个多媒体呈现。

[0080] ISO 基础媒体文件格式 [2,3,4] 可以有利地被采用作为本发明的媒体容器文件的文件格式。替换的容器文件格式包括 MP4 文件格式、3GP 文件格式以及 QuickTime 格式。

[0081] 异步分层编码 (ALC) 是一种显著可伸缩的可靠的内容递送协议。其是用于对任意二进制对象进行可靠的多点传送递送的基础协议, 并且已经被采用作为在 3GPP2BCMCS (广播 / 多点传送服务) 和开放移动联盟 (OMA) 浏览器和内容 (BAC) 广播 (BCAST) 工作组中进行广播 / 多点传送文件递送的强制协议。

[0082] FLUTE (通过单向传输的文件递送) 在 ALC 之上建立并且定义了一种用于单向文件递送的协议, 其近来已经在 3GPP MBMS 和 DVB-H IP 数据播送 (IPDC) 中被采用作为针对广播 / 多点传送文件递送的强制协议。ALC 和 FLUTE 都由因特网工程特别任务组 (IETF) 定义。

[0083] FLUTE 定义了文件递送表 (FDT), 其携带与在 ALC 会话中递送的文件相关联的元数据, 并且提供用于对 FDT 进行带内递送和更新的机制。与此相对, ALC 依赖于对文件元数据进行带外递送的其他措施。OMA BCAST 定义了电子服务指南 (ESG), 其通常在所述 ALC 会话之前很好被递送到客户端。如果在所述 ALC 会话期间需要更新所述文件元数据, 则可以通过利用 ESG 递送 / 更新信道来更新 ESG 的各片段。

[0084] 将要通过 ALC 或者 FLUTE 递送的文件可以作为项目被存储在 ISO 容器文件中。元框 (Meta box) 及其子框 (child box) 允许把多种数据项目 (比如静态媒体 (画面) 和 SMIL 呈现) 存储到 ISO 基础媒体文件中。所述元框及其子框还允许在所述 ISO 基础媒体文件中把文件名、到各项目的路径以及所述文件目录结构的信令相关联。

[0085] 一般来说, 在可以通过 ALC/FLUTE 发送文件之前的第一步是把所述文件分割成源块和源符号。此外, 根据本发明, 计算 FEC 编码的奇偶符号。所述分割可以依赖于 FEC 方案、目标分组尺寸以及所期望的 FEC 开销。对于将被 FEC 编码的每一个源块, 预先计算奇偶符号的存储库, 并且将其与关于所述 FEC 方案和对所述源文件的分割的信息一起存储在所述 ISO 基础媒体文件中。

[0086] 促进文件传输的下一步是令所述 ISO 基础媒体文件还包含用于多点传送 / 广播服务器的指令, 其 (利用会话描述协议) 描述所述 ALC/FLUTE 会话以及描述如何把各项目封装到 ALC 或 FLUTE 分组中。

[0087] 一方面所述文件分割和 FEC 存储库, 另一方面用于文件递送的所述提示轨道被彼此独立地使用。所述文件分割和 FEC 存储库有助于提示轨道的设计并且允许例如具有不同 FEC 开销的替换提示轨道以重复使用相同的 FEC 符号。所述文件分割和 FEC 存储库还提供针对后递送修复独立地访问源符号和附加 FEC 符号的措施, 其中所述后递送修复可以通过 ALC/FLUTE 执行或者通过另一种协议带外执行。然而, 为了在服务器遵循提示轨道指令时降低复杂度, 提示轨道直接涉及到被拷贝到提示采样中的项目或数据的数据范围。

[0088] 下面给出了根据本发明的容器文件的一个更加详细的实现方式实例, 其具有 ISO

基础媒体文件格式的形式并且被适配成通过 ALC/FLUTE 传输。然而这仅仅应当被视为本发明说明的一个说明性实例,在本发明的范围内可以对该实例做出明显的修改和改变。

[0089] 源文件和 FEC 存储库的存储

[0090] 预定通过 ALC/FLUTE 传输的文件被作为项目存储在充当容器文件的 ISO 基础媒体文件的顶级元框 (‘meta’) 中。项目位置框 (‘iloc’) 表明每一个项目 (媒体源文件) 在所述容器文件内的实际存储位置以及每一个项目的文件尺寸。每一个项目的文件名、内容类型 (MIME 类型) 等等由项目信息框 (‘iinf’) 提供。

[0091] 按照类似的方式,预先计算的 FEC 存储库被作为附加项目存储在所述元框中。如果源文件被分离成几个源块,则对于每一个源块的 FEC 存储库优选地被存储为单独的项目。FEC 存储库与原始源项目之间的关系被记录在下面章节中描述的文件递送 (FD) 项目信息框内。

[0092] FD 项目信息框

[0093] 关于分割源文件和 FEC 存储库的细节被提供在 FD 项目信息框 (‘fiin’) 中。该框优选地被用于采用了 FD 提示轨道的文件,并且优选地恰好有一个该框位于所述元框 (‘meta’) 内。该框被如下定义:

[0094] `aligned(8)class FDItemInformationBox extends FullBox(' fiin' ,version = 0,0)`

[0095] {

[0096] `unsigned int(16)entry count;`

[0097] `PartitionEntry[entry_count]partition entries;`

[0098] `SessionGroupBox session info;`

[0099] `GroupIdToNameBox group_id_to_name;`

[0100] }

[0101] 所述 FD 项目信息框中的每一个 PartitionEntry (分割条目) 提供关于对应于特定媒体源文件的特定文件分割、FEC 编码、相关联的 FEC 存储库以及元数据的细节。如果在所述 ISO 文件中使用了替换的 FEC 编码方案或者分割,则有可能为一个源文件提供多个条目。所有分割条目都可以被隐含地编号,并且第一条目的编号通常为 1。

[0102] 分割条目

[0103] 源的分割条目 (‘paen’) 被如下定义:

[0104] `aligned(8)class PartitionEntry extends Box(' paen' )`

[0105] {

[0106] `FilePartitionBox blocks_and_symbols;`

[0107] `FECReservoirBox FEC_symbol_locations;`

[0108] }

[0109] 该条目可以包含两个框,所述两个框一起提供关于如何对媒体源文件进行 FEC 编码的所有细节。

[0110] 文件分割框

[0111] 文件分割框 (‘fpar’) 识别出源文件并且提供把该文件分割成源块和符号。其定义如下:

```
[0112]   aligned(8)class FilePartitionBox extends FullBox(' fpar' ,version=0,
[0113]   {
[0114]       unsigned int(16)item_ID;
[0115]       unsigned int(16)packet_payload_size;
[0116]       unsigned int(16)FEC_encoding_ID;
[0117]       unsigned int(16)FEC_instance_ID;
[0118]       unsigned int(16)max_source_block_length;
[0119]       unsigned int(16)encoding_symbol_length;
[0120]       unsigned int(16)max_number_of_encoding_symbols;
[0121]       string scheme_specific_info;
[0122]       unsigned int(16)entry_count;
[0123]       for(i = 1;i <= entry_count;i++)
[0124]       {
[0125]           unsigned int(16)block_count;
[0126]           unsigned int(32)block_size;
[0127]       }
[0128]   }
```

[0129] 语义：

[0130] item\_ID(项目 ID)表明源文件的项目 ID。有可能通过在多于一个文件信息条目的文件分割框中使用相同的项目 ID 来提供对源文件的替换的分割和 / 或 FEC 编码。

[0131] packet\_payload\_size(分组有效载荷尺寸)给出分割算法的目标 FLUTE 或 ALC 分组有效载荷尺寸。应当注意到,UDP 分组有效载荷更大,这是因为其还包含 FLUTE 或 ALC 报头。

[0132] FEC\_encoding\_ID(FEC 编码 ID)标识出 FEC 编码方案。0 值可以对应于默认方案,比如对应于“Compact No-Code FEC scheme(紧致无代码 FEC 方案)”,其也被称作“Null-FEC(空 FEC)”[5]。1 值优选地对应于“MBMS FEC”[1]。

[0133] FEC\_instance\_ID(FEC 事例 ID)提供对被用于欠详列的 FEC 方案的 FEC 编码器的更为具体的标识。该值通常不被用于完全详列的 FEC 方案。关于欠详列的 FEC 方案的进一步细节参见文献 [5]。

[0134] max\_source\_block\_length(最大源块长度)给出每个媒体源块的最大源符号数目。

[0135] encoding\_symbol\_length(编码符号长度)给出一个编码符号(源符号和 FEC 奇偶符号)的(以字节计的)尺寸。除了可能较短的最后一个符号之外,一个项目的所有编码符号优选地具有相同的长度。

[0136] max\_number\_of\_encoding\_symbols(最大编码符号数目)给出针对在文献 [5] 中定义 FEC 编码 ID 129 可以为源块生成的最大编码符号数目。

[0137] scheme\_specific\_info(特定于方案的信息)是“FLUTEbis”中的特定于方案的对象传送信息(特定于 FEC-OTI 方案的信息)的 base64 编码的空字符结尾字符串。所述信

息的定义取决于所述 FEC 编码 ID。

[0138] entry\_count(条目计数)给出提供对源文件的分割的(block\_count, block\_size)对的列表中的条目数。从所述文件的开头开始,每一个条目表明该文件的下一个片段如何被划分成源块和源符号。

[0139] block\_count(块计数)表明尺寸为 block\_size(以字节计)的连续源块的数目。不是符号尺寸(其被提供在 FEC 信息框中)的倍数的 block\_size 表明最后一个源符号包括未被存储在所述文件项目中的填充符。

[0140] FEC 存储库框

[0141] FEC 存储库框('fecr')把媒体源文件与被存储为附加项目的 FEC 存储库相关联:

[0142] aligned(8)class FECReservoirBox extends FullBox(' fecr',version = 0,0)

```
[0143] {  
[0144]     unsigned int(16)entry_count;  
[0145]     for(i = 1;i <= entry_count;i++)  
[0146]     {  
[0147]         unsigned int(16)item_ID;  
[0148]         unsigned int(32)symbol_count;  
[0149]     }  
[0150] }
```

[0151] 语义:

[0152] entry\_count(条目计数)给出提供对于每一个 FEC 存储库及其所包含的源符号数目的(item\_ID, symbol\_count)对的列表中的条目数。该列表开始于与所述媒体源文件的第一个源块相关联的 FEC 存储库,并且顺序地继续遍历该文件。

[0153] 项目信息框

[0154] 为了利用广播/多点传送文件下载协议(ALC/FLUTE)内部地传送所嵌入的分立媒体,优选地使所述服务器还传送对应于所述分立媒体的某些元数据。如果 FLUTE 被用作广播协议,则作为所述 FDT 的一部分发送所述元数据,而如果把 ACL 与 OMA BCAST ESG 相结合地使用,则作为 OMABCAST ESG 的一部分发送所述元数据。

[0155] 由于某些元数据信息可能是快速(on the fly)产生的,因此对于 FLUTE 和 ALC 所共有的所述元数据的静态部分的模板结构被定义为所述项目信息条目的第二版本。所述项目信息条目的该版本在所述项目信息框中被用于具有源文件分割的项目。

[0156] aligned(8)class ItemInfoEntry extends FullBox(' infe',version = 1,0)

```
[0157] {  
[0158]     unsigned int(16)item ID;  
[0159]     unsigned int(16)item_protection_index;  
[0160]     unsigned int(32)content length;  
[0161]     unsigned int(32)transfer length;  
[0162]     string item_name;
```

```
[0163]         string content_type ;
[0164]         string content_location ;
[0165]         string content_encoding ;
[0166]         string content_MD5 ;
[0167]         unsigned int(8)entry_count ;
[0168]         for(i = 1 ;i <= entry_count ;i++)
[0169]         {
[0170]             unsigned int(32) group_id ;
[0171]         }
[0172]     }
```

[0173] 语义：

[0174] item\_id(项目 id) 对于主资源（例如包含在“xml”框中的扩展标记语言 (XML)）包含 0, 或者对于为之定义了以下信息的项目包含该项目的 ID。

[0175] item\_protection\_index(项目保护索引) 对于不受保护的项目包含 0, 或者包含到项目保护框的基于 1 的索引（所述项目保护框中的第一个框的索引为 1），其中所述项目保护框定义了被应用于该项目的保护。

[0176] content\_length(内容长度) 给出（未编码）文件的总长度（以字节计）。

[0177] transfer\_length(传送长度) 给出（已编码）文件的总长度（以字节计）。应当注意到, 如果没有应用内容编码则该传送长度等于内容长度（见下面）。

[0178] item\_name(项目名) 是 UTF-8 字符的空字符结尾字符串, 其包含所述项目的符号名, 即该项目（源文件）的文件名。

[0179] content\_type(内容类型) 是 UTF-8 字符的空字符结尾字符串, 其具有所述项目的 MIME 类型。如果该项目是经过内容编码的（见下面），则所述 MIME 类型指代内容解码之后的该项目。

[0180] content\_location(内容位置) 是 UTF-8 字符的空字符结尾字符串, 其包含定义在 HTTP/1.1[6] 中的所述文件的 URI。

[0181] content\_encoding(内容编码) 是 UTF-8 字符的空字符结尾字符串, 其被用来表明所述二进制文件是已编码的, 并且在被解译之前需要被解码。针对 HTTP/1.1 的内容编码定义了其值。一些可能的值有“gzip”、“compress”和“deflate”。空的字符串表明没有内容编码。应当注意到, 所述项目在应用了内容编码之后被存储。

[0182] content\_MD5(内容 MD5) 是 UTF-8 字符的空字符结尾字符串, 其包含所述文件的 MD5 摘要 [6, 7]。

[0183] entry\_count(条目计数) 给出下面的列表中的条目数。

[0184] group\_ID(组 ID) 表明所述文件项目所属的文件组。

[0185] 优选地采用所有的字段。但是空字符结尾字符串也有可能仅仅包含空字符, 以表明没有提供该字段的相应值。对所述框的未来扩展可以在末尾添加附加的字段。

[0186] 通过考虑被提供在对于每一个项目的文件信息框中的信息以及由提示轨道所使用的项目列表, 可以构造对于 FDT 或 ESG 所需的文件条目。

[0187] 可以通过使用定义在 ISO 基础媒体文件格式 [2, 3] 的 8.44.7 节中的通用资源位

置 (URL) 表来指代所嵌入的媒体资源的内容位置。

[0188] 会话组框

[0189] FD 会话可以同时几条 FD 信道上进行发送, 其中的每一条信道都由 FD 提示轨道进行描述。所述会话组框包含会话列表以及属于每一个会话的所有媒体文件组和提示轨道。如果在所述容器文件中有多于一条 FD 提示轨道, 则在所述 FD 项目信息框中优选地存在一个会话组框。

[0190] 在任意时间应当仅仅处理一个会话组。在会话组中第一条列出的提示轨道指定基础信道。如果所述媒体服务器在各会话组之间没有优选, 则默认的选择通常是第一会话组。包含由所述提示轨道所提到的文件的所有文件组的组 ID 都被包括在所述文件组的列表中。随后可以 (利用组 ID 到名称框) 把所述文件组 ID 翻译成文件组名, 所述文件组名可以被所述服务器包括在 FDT 中。

[0191] aligned(8)class SessionGroupBox extends Box(' segr' )

[0192] {

[0193] unsigned int(16)num\_session\_groups;

[0194] for(i = 0 ;i < num\_session\_groups ;i++)

[0195] {

[0196] unsigned int(8)entry\_count;

[0197] for (j = 0 ;j < entry\_count ;j++)

[0198] {

[0199] unsignedint(32)group\_ID;

[0200] }

[0201] unsigned int(16)num\_channels\_in\_session\_group;

[0202] for(k = 0 ;k < num\_channels\_in\_session\_group ;k++)

[0203] {

[0204] unsigned int(32)hint\_track\_id;

[0205] }

[0206] }

[0207] }

[0208] 语义:

[0209] num\_session\_groups(会话组数目)指定会话组的数目。

[0210] entry\_count(条目计数)给出下面的列表中的条目数, 该列表包括所述会话组所遵照的所有文件组。该会话组包含被包括在由每一个源文件的项目信息条目所表明的所列出的文件组中的所有文件。用于该会话组的 FDT 应当优选地仅仅包含在该结构中列出的那些组。

[0211] group\_ID(组 ID)表明所述会话组所遵照的文件组。

[0212] num\_channels\_in\_session\_groups(会话组中的信道数)指定所述会话组中的信道数。该会话组中的信道数的值是正整数。

[0213] hint\_track\_ID(提示轨道 ID)指定属于特定会话组的 FD 提示轨道的轨道 ID。一条 FD 提示轨道对应于一条分层编码传输 (LCT) 信道。

[0214] 组 ID 到名称框

[0215] 所述组 ID 到名称框把文件组名与使用在所述项目信息条目中的文件组 ID 相关联。

[0216] aligned(8)class GroupIdToNameBox extends FullBox( ' gitn' ,version = 0, 0)

[0217] {

[0218]        unsigned int(32)entry\_count ;

[0219]        for(i = 1 ;i <= entry\_count ;i++)

[0220]        {

[0221]            unsigned int(32)group\_ID ;

[0222]            string group\_name ;

[0223]        }

[0224] }

[0225] 语义：

[0226] entry\_count( 条目计数 ) 给出下面的列表中的条目数。

[0227] group\_ID( 组 ID) 表明文件组。

[0228] group\_name( 组名 ) 是 UTF-8 字符的空字符结尾字符串,其包含相应的文件组名。

[0229] 提示轨道格式

[0230] 所述提示轨道结构被一般化,以便支持多种数据格式的提示采样。所述提示轨道采样包含建立正确类型的分组报头所需要的任何数据,并且还包含到应归入所述分组中的数据的媒体源块的指针。

[0231] 采样条目格式

[0232] FD 提示轨道是具有“fdp”(“文件递送协议”的简称)的采样描述中的条目格式的提示轨道(媒体句柄“hint(提示)”)。FDHint SampleEntry(FD 提示采样条目)被包含在 SampleDescriptionBox( 采样描述框 )( ‘std’) 中并且具有以下句法：

[0233] class FDHintSampleEntry()extends SampleEntry( ‘fdp ‘)

[0234] {

[0235]        uint(16)hinttrackversion = 1 ;

[0236]        uint(16)highestcompatibleversion = 1 ;

[0237]        uint(16)partition\_entry\_ID ;

[0238]        uint(16)FEC\_overhead ;

[0239]        box additionaldata[] ;

[0240] }

[0241] 语义：

[0242] partition\_entry\_ID( 分割条目 ID) 表明 FD 项目信息框中的分割条目。0 值表明没有分割条目与该采样条目相关联(例如对于 FDT)。

[0243] FEC\_overhead(FEC 开销)是固定的值 8.8,其表明由所述一个(或多个)提示采样使用的百分比保护开销。提供 FEC 开销的意图是提供帮助媒体服务器选择会话组(以及相应的 FD 提示轨道)的特性。

[0244] 字段“hinttrackversion（提示轨道版本）”和“highestcompatibleversion（最高兼容版本）”具有与“RtpHintSampleEntry（Rtp 提示采样条目）”中相同的解译，这在 ISO 基础媒体文件格式 [2,3] 的 10.2 节中做了描述。作为附加的数据，提供 time\_scale\_entry（时间尺度条目）框。如果没有提供所述时间尺度条目框，则没有给出关于分组定时的指示。

[0245] 可以通过观察提示轨道的所有采样条目以及由上述项目 ID 所涉及的项目的相应文件元数据信息框来产生 FDT 或 ESG 所需要的文件条目。如果任何采样没有涉及采样条目，则没有采样条目包括在所述提示轨道中。

[0246] 推荐所述媒体服务器对于所述文件的每一次重传发送不同的一组 FEC 符号。

[0247] 采样格式

[0248] 所述提示轨道中的每一个 FD 采样将生成一个或多个 FD 分组。每一个采样包含两个区域：针对组成所述分组的指令，以及在发送所述分组时所需要的任何额外数据（例如被拷贝到所述采样中而不是驻留在用于源文件或 FEC 的项目中的编码符号）。应当注意到，可以从采样尺寸表中获知所述采样的尺寸。

[0249] aligned(8)class FDsample extends Box( 'fdsa' )

[0250] {

[0251]       FDPacketBox packetbox[]

[0252]       ExtraDataBox extradata;

[0253] }

[0254] FD 采样的采样号定义应当由所述媒体服务器对其进行处理的顺序。同样地，每一个 FD 采样中的 FD 分组框按照应当对其进行处理的顺序出现。如果在所述 FD 提示采样条目中存在时间尺度条目框，则定义采样次数并且提供对于默认比特率的相对分组发送次数。取决于实际的传输比特率，服务器可以应用线性时间缩放。采样次数可以简化所述调度处理，但是要及时发送分组则取决于所述媒体服务器。

[0255] 分组条目格式

[0256] 所述 FD 采样中的每一个分组具有以下结构 [8-10]：

[0257] aligned(8)class FDpacketBox extends Box( 'fdpa' )

[0258] {

[0259]       header\_template LCT\_header\_info;

[0260]       unsigned int(16)entrycount1;

[0261]       dataentry header\_extension\_constructors[entrycount1];

[0262]       unsigned int(16)entrycount2;

[0263]       dataentry packet\_constructors[entrycount2];

[0264] }

[0265] LCT\_header\_info(LCT 报头信息) 包含用于当前 FD 分组的 LCT 报头模板。

[0266] entry\_count1( 条目计数 1) 对下面的构造器进行计数。

[0267] header\_extension\_constructors:( 报头扩展构造器 ) 是被用来构造 LCT 报头扩展的结构。

[0268] entry\_count2( 条目计数 2) 对下面的构造器进行计数。

[0269] packet\_constructors(分组构造器)是被用来构造 FD 分组中的 FEC 有效载荷 ID 和源符号的结构。

[0270] LCT 报头模板格式

[0271] class header\_template

[0272] {

[0273] unsigned int(1)sender\_current\_time\_present;

[0274] unsigned int(1)expected\_residual\_time\_present;

[0275] unsigned int(1)session\_close\_bit;

[0276] unsigned int(1)object\_close\_bit;

[0277] unsigned int(4)reserved;

[0278] unsigned int(16)transport\_object\_identifier;

[0279] }

[0280] 所述 LCT 报头模板可以被媒体服务器使用来形成用于分组的 LCT 报头。应当注意到,所述报头的某些部分取决于服务器策略并且没有被包括在所述模板中。某些字段长度还取决于由所述服务器分配的 LCT 报头比特。所述服务器可能还需要改变所述 TOI 的值。

[0281] LCT 报头扩展构造器格式

[0282] 应当注意到,媒体服务器可以通过观察是否存在 EXT\_FDT 来识别出包括 FDT 的分组。

[0283] aligned(8)class LCTheaderextension

[0284] {

[0285] unsigned int(8)header\_extension\_type;

[0286] unsigned int(8)header\_extension\_length;

[0287] unsigned int(8)header\_extension\_content[];

[0288] }

[0289] header\_extension\_length(报头扩展长度)用 32 比特字的倍数表示。0 值意味着该报头由服务器生成。

[0290] header\_extension\_content(报头扩展内容)是等于 header\_extension\_length(报头扩展长度)的项目数。

[0291] 分组构造器格式

[0292] 存在多种形式的构造器。每一个构造器都是 16 字节,以便使得迭代更加容易。第一个字节是联合区分符(union discriminator)。该结构是基于 ISO 基础媒体文件格式 [2,3] 的 10.3.2 节。所述分组构造器被用来把 FEC 有效载荷 ID 以及源符号包括在 FD 分组中。

[0293] aligned(8)class FDconstructor(type)

[0294] {

[0295] unsigned int(8)constructor\_type = type;

[0296] }

[0297] aligned(8)class FDnoopconstructor extends FDconstructor(0)

[0298] {

```
[0299]         unsigned int(8)pad[15];
[0300]     }
[0301]     aligned(8)class FDimmediateconstructor extends FDconstructor(1)
[0302]     {
[0303]         unsigned int(8)count;
[0304]         unsigned int(8)data[count];
[0305]         unsigned int(8)pad[14 ~ count];
[0306]     }
[0307]     aligned(8)class FDsampleconstructor extends FDconstructor(2)
[0308]     {
[0309]         signed int(8)trackrefindex;
[0310]         unsigned int(16)length;
[0311]         unsigned int(32)samplenum;
[0312]         unsigned int(32)sampleoffset;
[0313]         unsigned int(16)bytesperblock = 1;
[0314]         unsigned int(16)samplesperblock = 1;
[0315]     }
[0316]     aligned(8)class FDitemconstructor extends FDconstructor(3)
[0317]     {
[0318]         unsigned int(16)item_ID;
[0319]         unsigned int(16)extent_index;
[0320]         unsigned int(64)data_offset;
[0321]         unsigned int(24)data_length;
[0322]     }
[0323]     aligned(8)class FDxmlboxconstructor extends FDconstructor(4)
[0324]     {
[0325]         unsigned int(64)data_offset;
[0326]         unsigned int(32)data_length;
[0327]         unsigned int(24)reserved;
[0328]     }
[0329]     额外数据框
[0330]     FD 提示轨道的每一个采样可以包括存储在额外数据框中的额外数据：
[0331]     aligned(8)class ExtraDataBox extends Box( 'extr' )
[0332]     {
[0333]         bit(8)extradata[];
[0334]     }
```

[0335] 图 5 是示出了根据本发明的媒体会话管理方法的流程图。该媒体会话管理是在媒体服务器（比如流传输或下载服务器）中实施的，并且其使用本发明的媒体容器文件。所述方法开始于步骤 S40，在该步骤中提供媒体容器文件。这一文件提供可以通过从所述媒体

服务器的存储器位置获取所述容器文件而实现,这意味着该服务器先前已经从内容提供商或产生器接收了所述文件。可替换地,所述媒体服务器可以结合针对媒体数据的请求从内容提供商订购或接收所述容器文件。

[0336] 在下一个步骤 S41 中,通过从所述容器文件的一个(或多个)媒体源块中提取媒体数据并且从一个(或多个)FEC 存储库中提取 FEC 冗余数据来编辑媒体数据分组。这一数据提取是基于所述容器文件中的元数据执行的,所述元数据提供所述一个(或多个)媒体源块与所述一个(或多个)FEC 存储库之间的关联。因此,媒体服务器优选地接收将在所述媒体会话期间传送的媒体数据的标识符。可替换地,所述容器文件可能仅仅包含单一媒体数据文件的媒体数据,因此没有必要选择媒体源。在任一种情况下,被包括在所述容器文件中(比如被包括在所述文件属性表中)的先前描述的信息都可以被用于识别所述媒体文件的开头,即应当从该处开始传输的第一个媒体源块。此外,被包括在所述容器文件中的其他信息可以被用作关于应当如何组合媒体数据和 FEC 数据并且将其包括在数据分组中的指令,其中所述数据分组适于通过一条或多条基于无线电的信道被无线传送到不同的客户端。在给出当前从中提取媒体数据的媒体源块的情况下,本发明的元数据允许识别出应当在所述分组编辑步骤 S41 中从其中提取 FEC 数据的相关联的 FEC 存储库。

[0337] 在下一个步骤 S42 中,优选地通过广播或多点传送技术把具有 FEC 可靠性保护的所编辑的媒体数据分组发送到客户端,在所述客户端处可以呈现所述媒体数据。通常一旦在所述媒体服务器中的传送缓冲器达到了给定水平之后就启动所述分组传输。但是在所述媒体会话期间,在其他分组正被传送的同时编辑新的数据分组并且将其输入到所述传送缓冲器中,这由线 L2 示意性地示出。

[0338] 所生成的容器文件以及在其中组织媒体数据并且提供预先计算的 FEC 数据,可以在媒体会话期间降低所述媒体服务器的处理需求。因此这导致降低了服务器复杂度并且允许服务器灵活性,这是因为所述服务器不需要快速进行源块构造和 FEC 编码。与此相对,所述服务器使用所述容器文件中的元数据和指令来提取预先计算的源符号和 FEC 符号、添加报头信息并且把所得到的数据分组发送到客户端。

[0339] 所述方法随后结束。

[0340] 图 6 是示出了图 4 的会话管理方法的附加步骤的流程图。所述方法从图 5 的步骤 S40 继续。在下一个步骤 S50 中确定当前可以被采用来在所述媒体会话中进行数据传输的 FEC 开销容量。可以基于针对所述媒体传输所能分配给所述服务器的带宽水平以及对于该媒体传输所采用的一个(或多个)无线电载体的最小和最大比特率水平等等来确定或者至少估计所述容量。实际上可以在该步骤 S50 中采用本领域中已知的用于结合数据传输确定这种开销容量的任何技术。

[0341] 一旦确定了所述 FEC 开销容量,下一个步骤 S51 就基于所确定的开销容量来选择编辑指令集。因此,所述媒体容器文件于是包含多个替换的编辑指令集,所述替换的编辑指令集可以被用于给定的媒体内容但是提供不同的 FEC 开销水平。换句话说,这些替换的编辑指令基本上定义将在编辑媒体数据分组时添加到所述媒体数据中的 FEC 冗余数据量。可接受的 FEC 开销越大,所添加的 FEC 数据就越多。通过具有不同的替换编辑指令,所述媒体服务器可以使用在给定当前开销限制的情况下允许最高可允许 FEC 保护的那些指令,从而与使用单一编辑指令集相比增大了在不同客户端处成功接收及解码所述媒体数据的机会。

[0342] 所述方法随后继续到图 5 的步骤 S41,在该步骤中基于在步骤 S51 中选择的编辑指令从所述媒体内容数据和相关联的 FEC 数据编辑媒体数据分组。

[0343] 图 12 是根据本发明的媒体容器文件 1 的示意图,其被用于显示出根据本发明的一个实施例使用替换的编辑指令。该容器文件 1 包括媒体源文件 10,所述媒体源文件 10 优选地包括多个媒体源块(比如两个媒体源块)。在该实施例中,该源文件 10 的每一个媒体源块具有相关联的 FEC 存储库 30、32,所述 FEC 存储库包括为对应的源块预先计算的 FEC 冗余数据。在该说明性例子中,该容器文件 1 还包括三条提示轨道 50、52、54,所述提示轨道包含用于不同的 FEC 开销的编辑指令。例如,在期望 10% 的冗余开销时可以使用第一提示轨道 50,第二提示轨道 52 给出大约 12% 的 FEC 开销,第三提示轨道 54 给出 14% 的 FEC 开销。在该图中采用了在文献 [1] 的附录 B 中提出的源块构造算法。如果选择了第一提示轨道 50,则生成数据分组 81、82、83、84 的第一流(在该图中每个媒体源块仅仅一个数据分组并且示出了 FEC 块)。然而,如果替换地使用了第二提示轨道 52,则生成数据分组 91、92、93、94 的第二流。与第一流 80 相比,第二流 90 对于每个媒体源块包括更大的 FEC 块,即更多的 FEC 冗余数据。但是对应的源块在两个流 80、90 中包含相同的媒体数据量。

[0344] 如前所述,所述媒体容器文件可以包含多个媒体源文件或者携带不同媒体内容的文件。图 7 的流程图示出了其中利用了这种多文件解决方案的图 5 的编辑和传送步骤的实施例。所述方法从图 5 的步骤 S40 继续。在下一个步骤 S60 中,第一媒体内容文件和第一 FEC 存储库的媒体数据分组基于所述元数据以及专用于该特定媒体内容的编辑指令而生成。所述分组生成的结果可以被视为包含第一组的媒体内容和来自第一 FEC 存储库的 FEC 数据的数据分组流。随后在步骤 S61 中利用基于无线电的通信信道(优选地是广播或多点传送信道)把该第一组数据分组发送到发出请求的客户端。

[0345] 然而在当前媒体会话中还应当把所述容器文件中的第二媒体源文件或者第二组的媒体内容传送到所述客户端。结果,在步骤 S62 中包含来自所述第二媒体源文件的媒体内容数据和来自第二相关联的 FEC 存储库的 FEC 冗余数据的数据分组基于元数据和与这个内容相关联的编辑指令而生成。随后在步骤 S63 中利用在先前描述的步骤 S61 中被所述服务器所采用的相同的基于无线电的通信信道来传送所得到的数据分组。因此,将作为数据分组的连续流来发送所述两组数据分组。

[0346] 图 8 的流程图示出了其中利用了多文件解决方案的图 5 的编辑和传送步骤的附加实施例。该方法从图 5 的步骤 40 继续。在下一个步骤 S70 中,生成包含来自第一媒体源文件的媒体内容数据和来自第一 FEC 存储库的 FEC 数据的媒体数据分组。该步骤 S70 基本上对应于图 7 的步骤 S60,并且在这里不再进一步描述。下一个步骤 S71 对应于图 6 的步骤 S61,在该步骤中数据分组包括从所述容器文件的第二媒体源文件和第二 FEC 存储库导出的媒体内容。在该实施例中,所述媒体服务器可以同时管理两个客户端组(IP 多点传送组)。因此,在下一步骤 S72 中,所述媒体服务器利用第一基于无线电的通信信道把在步骤 S70 中生成的数据分组的第一组或第一流传送到第一媒体组的客户端成员,并且同时地或者至少部分同时地利用第二基于无线电的通信信道把第二媒体数据分组流传送到第二媒体组的客户端成员。因此,在该实施例中将作为并行媒体流来发送所述两组数据分组。所述方法随后结束。

[0347] 本发明还包含以下情况:多个媒体文件的媒体数据被共同编辑并传送到一组客户

端。在这种情况下,在所述媒体会话期间由所述媒体服务器一同管理来自所述多个源文件的媒体数据。这特别是其中一个媒体文件包含视频数据并且第二媒体文件包含相关联的音频数据的情况。可以利用相同的基于无线电的通信信道或者不同的此类信道来传送所述两个文件的媒体数据。

[0348] 本发明预期在上面结合图 7 和 8 描述的教导可以被扩展成处理在所述容器文件中包括多于两个媒体源文件和 FEC 存储库的情况。

[0349] 图 13 是媒体容器文件 1 的示意性表示,其具有多个媒体源文件 10、12、14 和多个 FEC 存储库 30、32、34。该容器文件 1 还包括两条提示轨道 50、52,其具有用于从所述源文件 10、12、14 中提取媒体内容数据并且从所述存储库 30、32、34 中提取 FEC 数据所需要的编辑指令和元数据。

[0350] 在组成包含来自多个源文件 10、12、14 和 FEC 存储库 30、32、34 的数据分组 81、82、91、92 的流时可以由媒体服务器采用第一提示轨道 50。随后将利用基于无线电的信道把所生成的流传送到发出请求的客户端(与图 7 相比)。在应当同时管理多个媒体组时可以采用第二提示轨道 52。因此,由所述媒体服务器基于所述提示轨道 52、源文件 10、12、14 以及 FEC 存储库 30、32、34 生成数据分组 81、82 ;91、92 的多个并行流。

[0351] 如果所述媒体容器文件还包括附加信息(比如文件到块划分的信息、块分割的信息、FEC 算法和/或文件属性表的信息),所述媒体服务器可以在数据分组生成和传输过程中使用该附加信息。

[0352] 例如,所述媒体服务器可以把所述块分割的信息与所述元数据一起使用来从所述容器文件内的媒体源文件和块中提取媒体数据。这样,所述分割信息帮助该服务器正确地识别出所期望的媒体数据在所述容器文件中的正确的存储位置。相应地,所述媒体服务器可以把 FEC 算法信息与所述元数据一起使用来从所述容器文件中提取媒体数据和 FEC 数据。不同的 FEC 算法需要从媒体内容到源符号和块的不同分割,此外,所述分割可以取决于所述保护开销。因此,在预先计算所述 FEC 数据时所采用的 FEC 算法和其他 FEC 参数的信息对所述媒体服务器来说可能是有用的。

[0353] 可以在所述文件属性表中包括或者至少声明可以由所述媒体服务器使用的所述附加数据以及优选地还有 MIME 类型的信息、任何编码信息、尺寸信息等等。在一种优选实现方式中,该属性表构成单一信息或查找源,其可以被所述媒体服务器访问以便获得结合媒体提取、数据分组编辑和传输所需要的或者有利的信息。

[0354] 本发明的媒体服务器还可以被用于后修复程序。因此,在这种情况下,尽管在所述媒体流中包括了 FEC 冗余数据,但是在把媒体内容数据和 FEC 数据传送(多点传送)到客户端以后,某些客户端可能仍无法正确地解码所接收的媒体符号。图 9 的流程图示出了定义这种后修复程序的附加步骤。通常一旦在所述媒体会话期间的媒体传输结束或者至少在把某些媒体数据传送到客户端之后启动该程序。所述方法因此从图 5 的步骤 S42 继续。在下一个步骤 S80 中,所述媒体服务器接收到源自先前在与该服务器的媒体会话中所涉及到的客户端的针对后会话修复程序的请求。所述修复请求通常包括不正确地接收的媒体数据的标识符。该标识可以是特定媒体内容的名称并且可能是允许所述内容服务器更加详细地识别出未被所述客户端成功接收的内容部分的某种信息。在下一个步骤 S81 中,所述服务器使用该信息从基于该信息识别出的 FEC 存储库中提取 FEC 冗余数据。在该提取步骤 S81

中,所述服务器可以默认地提取在步骤 S82 中优选地利用基于单点传送的数据传输被传送到发出请求的客户端的预定义数目的 FEC 符号。如果所提供的 FEC 符号不足以对所述媒体内容进行成功解码,则所述客户端可以从所述服务器请求更多 FEC 符号。可替换地,所述客户端为所述服务器提供这样的信息,该信息允许该服务器至少估计出为了在该客户端处成功解码所述媒体内容所需要的 FEC 符号的数目。在这种情况下,所述服务器优选地在所述提取步骤 S81 中使用该信息,并且在步骤 S82 中把所提取的 FEC 符号传送到所述客户端。

[0355] 在媒体会话中使用了本发明的媒体容器文件之后,还可以在专用的后会话修复程序中使用所述媒体容器文件。图 10 是示出了本发明的这种后会话修复方法的流程图。该方法开始于步骤 S90,在该步骤中,修复服务器从客户端接收到针对 FEC 冗余数据的请求,该客户端先前在媒体会话中被涉及到并且从媒体服务器接收了媒体数据。在步骤 S90 中接收到的该请求包括允许该修复服务器识别出媒体容器文件的标识符。因此,该服务器在下一个步骤 S91 中基于所述请求(通常是该请求中的所述标识符)提供相关的媒体容器文件。该容器文件通常是由媒体服务器在先前的媒体会话中使用的容器文件的副本,因此包括一个(或多个)媒体源块和一个(或多个)FEC 存储库,所述媒体源块和 FEC 存储库分别包含所传送的但是没有在发出请求的客户端处完全成功接收到的媒体和 FEC 数据。可以从内容提供商/产生器或者实际从实施先前的媒体会话的所述媒体服务器订购或请求所述容器文件。

[0356] 所述修复服务器在步骤 S92 中基于元数据使用所述修复请求中的标识符从所述媒体容器文件中提取 FEC 冗余数据。在一种优选的实现方式中,所述标识符可以是包含所述不正确地接收的媒体内容的媒体源文件的名称或者某种更为详细的信息,其中包括特定媒体源块的标识,所述特定媒体源块包含所述缺失的媒体数据。随后把所述容器文件中的元数据与所述标识符一起使用来识别该容器文件中的至少一个 FEC 存储库,其中应当从该至少一个 FEC 存储库中提取所述 FEC 数据。如前面结合图 9 的步骤 S81 所描述的那样,可以向所述修复服务器通知对进行成功的数据解码所需的 FEC 数据量的估计。可替换地,所述修复服务器可以提取并发送默认数目的 FEC 符号,并且要求所述客户端在这些默认数目的符号不足够的情况下请求更多 FEC 数据。

[0357] 参见图 11,在一种优选的实现方式中,所述容器文件包括后会话修复指令 70。这些指令随后被所述修复服务器使用来编辑数据分组,所述数据分组包含从 FEC 存储库 30、32、34 中提取的 FEC 数据。如图所示,由所述服务器在该提取程序中使用的元数据 45 可以被包括在所述编辑指令 70 中或者构成其一部分。可替换地,可以与相关的媒体源文件 10、12、14、FEC 存储库 30、32、34 相结合地提供所述元数据 45,或者在被包括在所述容器文件 1 内的文件属性表 60 中提供所述元数据 45。

[0358] 图 14 是通信网络的示意性总览,其中示出了生成或使用本发明的媒体容器文件 1 的各方。内容服务器 100 表示所述内容提供商或产生器,其接收或者可以访问媒体源数据并且构造媒体容器文件 1。该容器文件 1 的副本被发送到媒体服务器 200,该媒体服务器 200 在媒体会话中使用该容器文件 1 来编辑数据分组,所述数据分组包含被发送(多点传送)到不同的客户端 400、410、420 的媒体和 FEC 数据,所述客户端在该图中由移动终端表示。在与该媒体服务器 200 的所述媒体会话之后,其中一个所述移动终端 400 可以联系修复服务器 300。在这种情况下,该修复服务器 300 从该内容服务器 100 请求所述媒体容器文

件 1 的副本,并且在后会话修复程序中使用该副本。

[0359] 图 15 是根据本发明的媒体内容服务器 100 的示意性方框图。该内容服务器 100 包括一般的输入和输出 (I/O) 单元 110,其被设置成与各外部单元通信并且包括用于与各外部单元通信的功能(发送器/接收器、调制器/解调器、编码器/解码器)。该 I/O 单元 110 特别被设置成接收输入媒体内容以及接收针对媒体容器文件的请求。当向所述通信网络中的其他服务器传送这种容器文件时,服务器 100 还采用该 I/O 单元 110。

[0360] 所述内容服务器 100 还包括容器文件产生器 120,其被设置成产生本发明的媒体容器文件。该产生器 120 包括媒体块管理器 130,其被设置成把至少一个媒体源块输入并组织到媒体容器文件中。该至少一个输入源块优选地是从该内容服务器 100 的文件划分器 190 提供的。可替换地,所述至少一个源块可以从内部数据存储装置 115 检索并且可以利用所述 I/O 单元 110 从外部媒体源 500、510 提供。

[0361] 所述内容服务器 100 的 FEC 编解码器 160 为由所述块管理器 130 输入到所述容器文件中的媒体源块计算 FEC 冗余数据。该 FEC 编解码器 160 可以在该计算程序中使用任一种先前提到的 FEC 算法。该编解码器 160 可以被设置成对于所有其冗余数据计算都使用给定的 FEC 算法。可替换地,该编解码器 160 可以使用多种不同算法,因此可以选择实际的算法。可以基于不同的参数来执行所述算法选择,所述参数比如是由所述文件划分器 190 进行的特定块划分、所述媒体源块中的数据类型或者某种其他参数。由所述编解码器 160 计算的 FEC 数据既可以被用作媒体会话中的冗余数据,也可以被用作后会话修复程序中的修复数据。

[0362] 所得到的计算出的 FEC 数据(符号)被输入到所述文件产生器 120 的 FEC 数据管理器 140。该 FEC 管理器 140 把所述输入 FEC 数据组织到所述媒体容器文件内的至少一个 FEC 存储库中。在一种优选的实现方式中,所述 FEC 编解码器 160 和 FEC 管理器 140 生成用于由所述块管理器 130 组织到所述文件中的每一个源块的 FEC 符号,并且在所述容器文件中为每一个源块提供至少一个 FEC 存储库。

[0363] 所述文件管理器 120 的元数据管理器 150 把元数据提供到所述容器文件中。该元数据提供由所述块管理器 130 组织的所述媒体源块与由所述 FEC 管理器 140 组织的所述 FEC 存储库之间的关联。

[0364] 随后可以把所得到的媒体容器文件至少临时存储在所述数据存储装置 115 中,或者通过所述 I/O 单元 110 将其传送到媒体服务器或修复服务器。

[0365] 在一种优选的实现方式中,所述输入媒体内容具有媒体源文件的形式,所述媒体源文件被提供到所述文件划分器 190。该划分器 190 把所述源文件分离成一个或多个源块。该划分器 190 可以基于不同的信息或参数进行该文件划分。例如,可以至少部分地基于由所述 FEC 编解码器 160 采用的 FEC 算法来确定所述文件划分。在这种情况下,所述文件划分 190 优选地可以使用这种 FEC 算法的信息。该划分器 190 随后可以把媒体源文件划分成 N-1 个等尺寸的媒体源块以及一个可以具有小于其他 N-1 个块的尺寸的媒体源块。

[0366] 所述媒体文件还被输入到块分割器 195,其优选地在所述文件划分器对所述数据进行了处理之后对所述媒体数据进行操作。该分割器 195 及时对一个媒体源块进行操作,并且把该块分割成具有已定义尺寸的源符号。优选地基于将被所述 FEC 编解码器 160 采用的特定 FEC 算法或方案来进行该块分割。所述块分割器 195 还可以操作来执行适于把所述

源符号置于数据分组中的分割,所述数据分组将由媒体服务器在媒体会话期间采用。因此,所述分割器 195 可以采用分组尺寸(比如 UDP 分组尺寸)信息。

[0367] 所述分割器 195 还生成所得到的块分割的信息。该信息随后被转送到所述 FEC 编解码器 160,以便在生成所述 FEC 冗余数据时使用。

[0368] 所述内容服务器 100 的各单元 110、120、130、140、150、160、190 和 195 可以被实现或者提供为软件、硬件或其组合。所述单元 110 到 195 都可以被实现在通信系统中的单一网络节点内的内容服务器 100 中。可替换地,分布式实现方式也是可能的,并且落在本发明的范围内。在这种情况下,所述内容服务器 100 的不同单元 110 到 195 可以被设置在不同的网络节点中,但是即便如此也仍将执行前面所描述的其预定操作。

[0369] 图 16 是更加详细地示出了图 15 的容器文件产生器 120 的实施例的示意性方框图。该文件产生器 120 还包括信息管理器 170,其用于把不同的信息数据包括并组织在所述容器文件中。该管理器 170 优选地把由所述 FEC 编解码器 160 使用的 FEC 算法、由所述文件划分器 190 使用的文件划分和/或所述块分割器 195 的块分割的信息提供在所述文件中。其他信息可以是描述所述媒体数据的数据,比如媒体类型、(块和/或符号)尺寸信息、内容名称或标识符、位置信息、可选的编码信息等等。在一个优选实施例中,可能把该信息与来自所述元管理器 150 的元数据一起组织到被包括在所述容器文件内的文件属性表中。

[0370] 所述文件产生器 120 的指令管理器 180 生成编辑指令并且将其插入到所述容器文件中。这些指令包括由媒体服务器使用来基于来自所述元数据管理器 150 的元数据编辑来自所述媒体源块的媒体数据以及来自所述 FEC 存储库的 FEC 数据的信息。该管理器 180 可以为所述文件中的每一项媒体内容生成单一指令或者指令集。可替换地,可以由该管理器 180 提供不同的所述指令并且将其组织到所述容器文件中,其中所述不同的指令适应于在所述媒体会话中采用的不同的 FEC 开销、不同的 FEC 数据类型和/或不同数目的基于无线电的通信信道。

[0371] 所述容器文件产生器 120 的各单元 130 到 180 可以被实现或者提供为软件、硬件或其组合。所述单元 130 到 180 都可以被实现在所述容器文件产生器 120 中。可替换地,分布式实现方式也是可能的,并且落在本发明的范围内。在这种情况下,所述容器文件产生器 120 的不同单元 120 到 180 可以被设置在所述内容服务器 100 中的其他位置处。

[0372] 图 17 是根据本发明的媒体会话服务器 200 的示意性方框图。该媒体服务器 200 包括 I/O 单元 210,其用于实施与各外部单元的通信。该 I/O 单元 210 特别被设置成从内容服务器请求并接收媒体文件容器。该 I/O 单元 210 还接收源自不同用户客户端的针对媒体内容的请求或者至少关于应当向哪些客户端传送媒体内容的信息。还可以通过该 I/O 单元 210 把由所述媒体服务器 200 编辑的数据分组传送到这些客户端。

[0373] 所述服务器 200 包括媒体文件提供器 220,其提供媒体内容文件以便使用在当前会话中。该文件提供器 220 可以生成针对特定容器文件的请求,该请求通过所述 I/O 单元 210 被传送到内容产生器。可替换地,该提供器 220 从被提供在所述媒体服务器 200 内的数据存储装置 260 中获取先前接收的容器文件。

[0374] 数据分组编辑器 230 使用被包括在来自所述提供器 220 的容器文件中的元数据以及优选地编辑指令以从该文件中提取媒体数据和 FEC 数据,并且生成包含所提取的该数据的数据分组。随后通过所述 I/O 单元 210 传送(或者从该处流传输或下载)如此生成的数

据分组。

[0375] 所述数据分组编辑器 230 在所述编辑处理中优选地还使用被包括在所述文件中的其他信息 (FEC 算法信息、划分 / 分割信息、尺寸信息、内容名称、内容存储信息)。因此, 实际上在所述内容文件中提供了生成具有媒体数据和 FEC 数据的数据分组所需要的所有指令和数据, 从而允许灵活且高效的媒体会话管理。

[0376] 对于给定的媒体内容可以把不同的编辑指令存储在所述文件中。例如, 所述指令可以是依赖于信道或者依赖于容量的。在前一种情况下, 可用无线电信道的数目和应当传送的并行媒体流的数目决定由所述编辑器 230 使用的实际编辑指令。在后一种情况下, 在所述服务器 200 中优选地包括 FEC 容量估计器 240, 其用于估计可以在会话期间采用的 FEC 开销的最大数量。优选地在所述会话期间动态地更新由该估计器 240 执行的开销估计, 这是因为可以在所述会话期间改变所述开销容量。集合选择器 250 使用来自所述估计器 240 的所述容量估计来选择将要使用可以在所述文件中获得的哪一条特定编辑指令或指令集。所述分组编辑器 230 随后使用该指令 (集) 把媒体数据和 FEC 数据编辑到数据分组中。

[0377] 所述媒体服务器 200 可以可选地包括 FEC 管理器 270, 其可以由该媒体服务器 200 在后会话修复程序中使用。在这种情况下, 所述 I/O 单元 210 接收源自客户端的修复请求, 其中该服务器 200 先前向该客户端传送了媒体和 FEC 数据。从所述 I/O 单元 210 把该请求转送到所述 FEC 管理器 270。该管理器 270 使用该请求从所述媒体容器文件中 (例如其被存储在所述数据存储装置 260 中) 识别并提取出 FEC 冗余数据 (符号)。在该后会话 FEC 提取中, 该 FEC 管理器 270 可以采用被包括在所述容器文件中的后会话修复指令来识别出正确的 FEC 数据以及 / 或者把所提取出的 FEC 数据编辑到数据分组中。随后通过所述 I/O 单元 210 把所得到的数据分组传送到发出请求的该客户端。

[0378] 所述媒体服务器 200 的各单元 210、220、230、240、250 和 270 可以被实现或者提供为软件、硬件或其组合。所述单元 210 到 270 都可以被实现在通信系统中的单一网络节点内的媒体服务器 200 中。可替换地, 分布式实现方式也是可能的, 并且落在本发明的范围内。在这种情况下, 所述媒体服务器 200 的不同单元 210 到 270 可以被设置在不同的网络节点中, 但是即便如此也仍将执行前面所描述的其预定操作。

[0379] 图 18 是根据本发明的修复服务器 300 的示意性方框图。该服务器 300 包括 I/O 单元 310, 其被设置成与各外部单元进行通信。该 I/O 单元 310 特别被提供来接收来自客户端的后会话修复请求, 并且用于从内容产生器或者媒体会话服务器请求媒体容器文件。对包括 FEC 冗余数据的数据分组的传输也通过该 I/O 单元 310 执行。

[0380] 所述修复服务器 300 包括媒体文件提供器 320, 其响应于接收到来自客户端的修复请求而提供容器文件。该提供器 320 可以组成通过所述 I/O 单元 310 传送到内容产生器的容器请求以提供所述文件。可替换地, 该修复服务器 300 先前已经接收了所述容器文件, 从而该文件提供器 320 从数据存储装置 340 中获取该容器文件。

[0381] FEC 数据提取器 330 被设置在所述服务器 300 中, 以便基于被包括在所述修复请求中的标识符从由所述文件提供器 320 提供的容器文件中提取 FEC 冗余数据。该标识符可以是未被所述客户端成功接收的媒体内容的名称或其他标识符。也可以使用其他标识符 (比如所述媒体会话服务器的标识符和 / 或所述媒体内容的递送时间), 其允许所述修复服务器 300 识别出正确的媒体内容 (有可能借助于到所述媒体服务器的请求)。还可以替换地

或附加地采用更加详细的信息,比如所述媒体内容的特定部分(其允许识别错误地接收的媒体源块)。

[0382] 由所述数据提取器 330 提取的 FEC 数据的数量可以是预先定义的。如果需要多于该预先定义的默认水平的 FEC 符号,则所述客户端必须相应地明确通知所述服务器 300,或者所述客户端需要发送针对另外的 FEC 数据的请求。可替换地,该提取器 330 从所述客户端接收关于为了成功解码所有媒体内容应当接收多少 FEC 符号的估计。该提取器 330 随后将在所述 FEC 数据提取中使用该估计。

[0383] 在一种优选的实现方式中,所述容器文件包括后会话修复指令,其定义把来自所述容器文件中的 FEC 存储库的 FEC 数据编辑到可以被发送给所述客户端的数据分组中的所述编辑。在这种情况下,所述提取器在所述提取和编辑处理中使用这些指令。

[0384] 所述修复服务器 300 的各单元 310 到 330 可以被实现或者提供为软件、硬件或其组合。所述单元 310 到 340 都可以被实现在通信系统中的单一网络节点内的修复服务器 300 中。可替换地,分布式实现方式也是可能的,并且落在本发明的范围内。在这种情况下,所述修复服务器 300 的不同单元 310 到 340 可以被设置在不同的网络节点中,但是即便如此也仍将执行前面所描述的其预定操作。

[0385] 本领域技术人员应当理解,在不偏离本发明的范围的情况下可以对本发明做出许多修改和改变,本发明的范围由所附权利要求书限定。

[0386] 参考文献

[0387] 3GPP TS 26.346 V7.0.0, 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Multimedia Broadcast/Multicast Service (MBMS); Protocols and codecs, June 2006

[0388] ISO/IEC 14496-12:2005:“ISO base media file format”

[0389] ISO/IEC 15444-12:2005:“ISO base media file format”

[0390] International application WO 2005/039131

[0391] RFC 3695; Compact Forward Error Correction (FEC) Schemes, February 2004

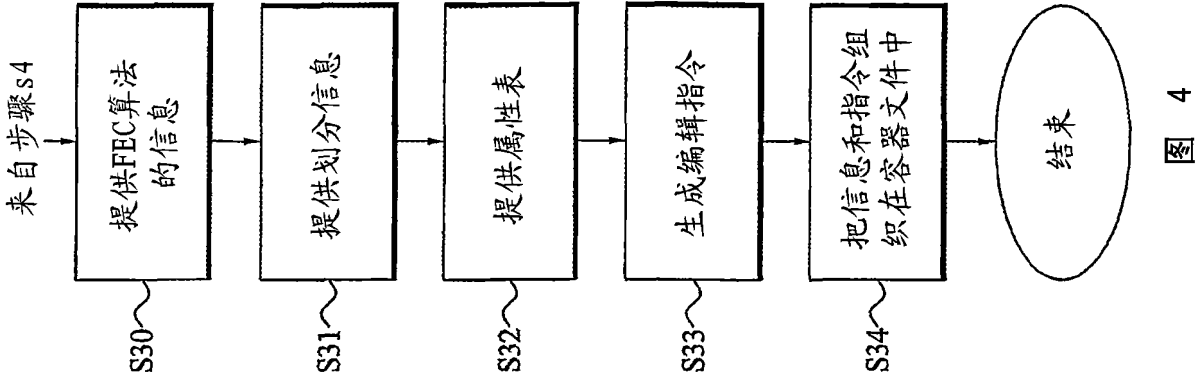
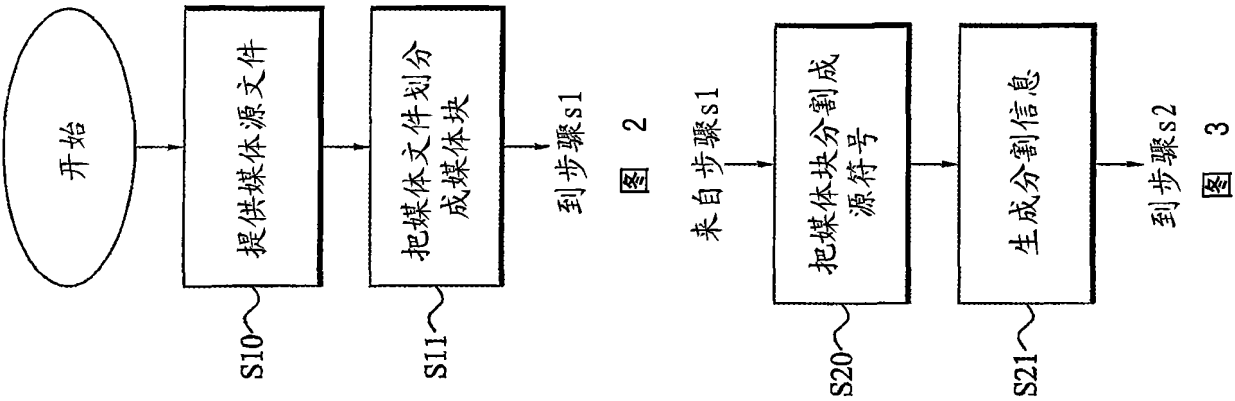
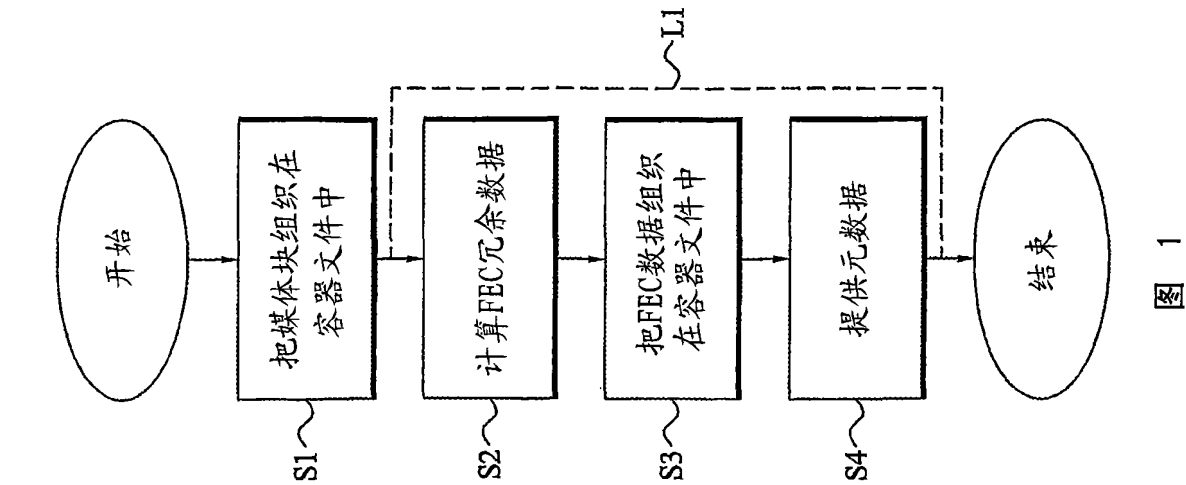
[0392] RFC 2616; Hypertext Transfer Protocol-HTTP/1.1, June 1999

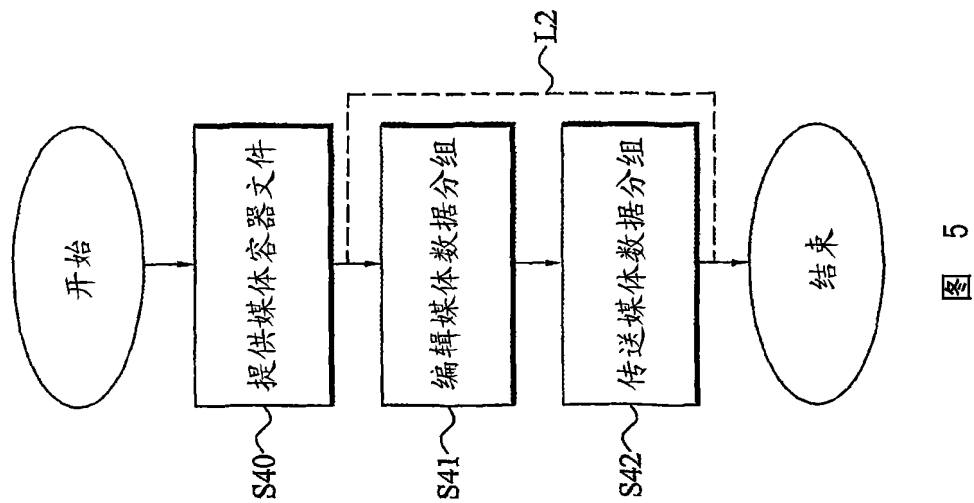
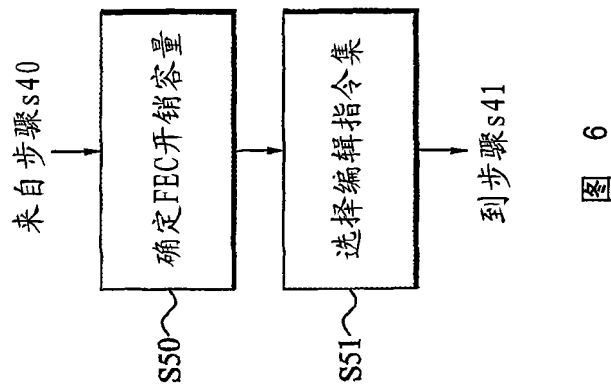
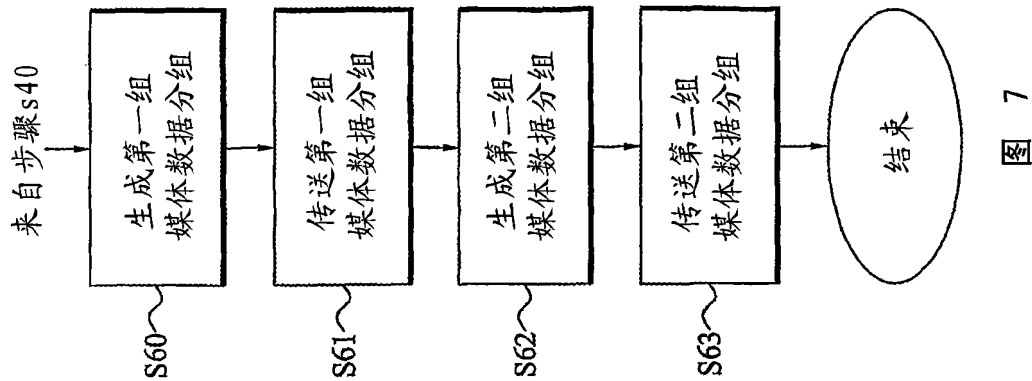
[0393] RFC 1864; The Content-MD5 Header Field, October 1995

[0394] RFC 3926; FLUTE-File Delivery over Unidirectional Transport, October 2004

[0395] [9] RFC 3450; Asynchronous Layered Coding (ALC) Protocol Instantiation, December 2002

[0396] [10] RFC 3451; Layered Coding Transport (LCT) Building Block, December 2002





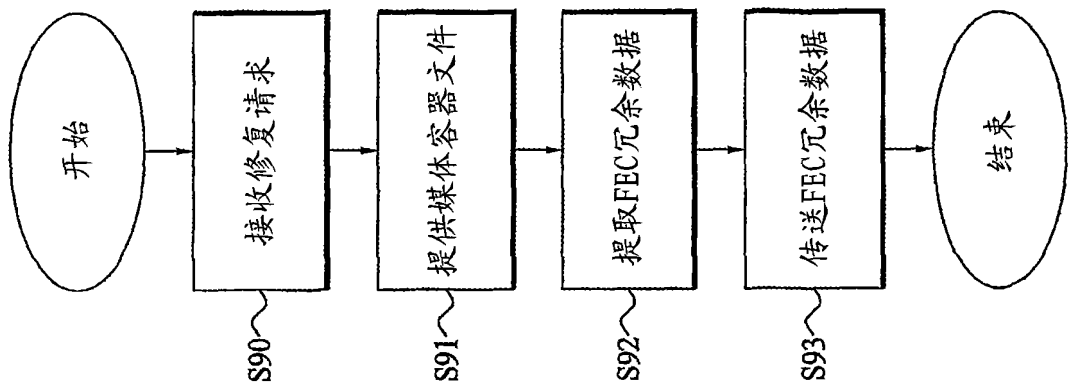


图 10

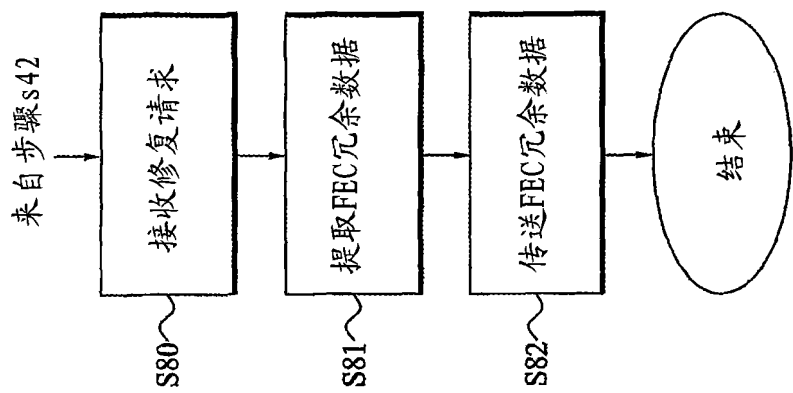


图 9

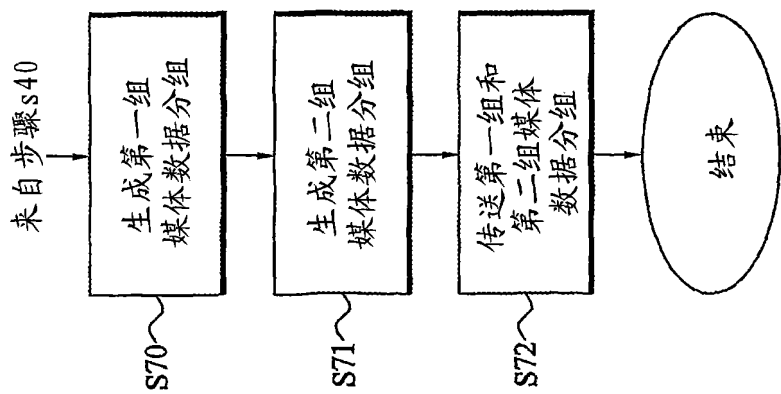


图 8

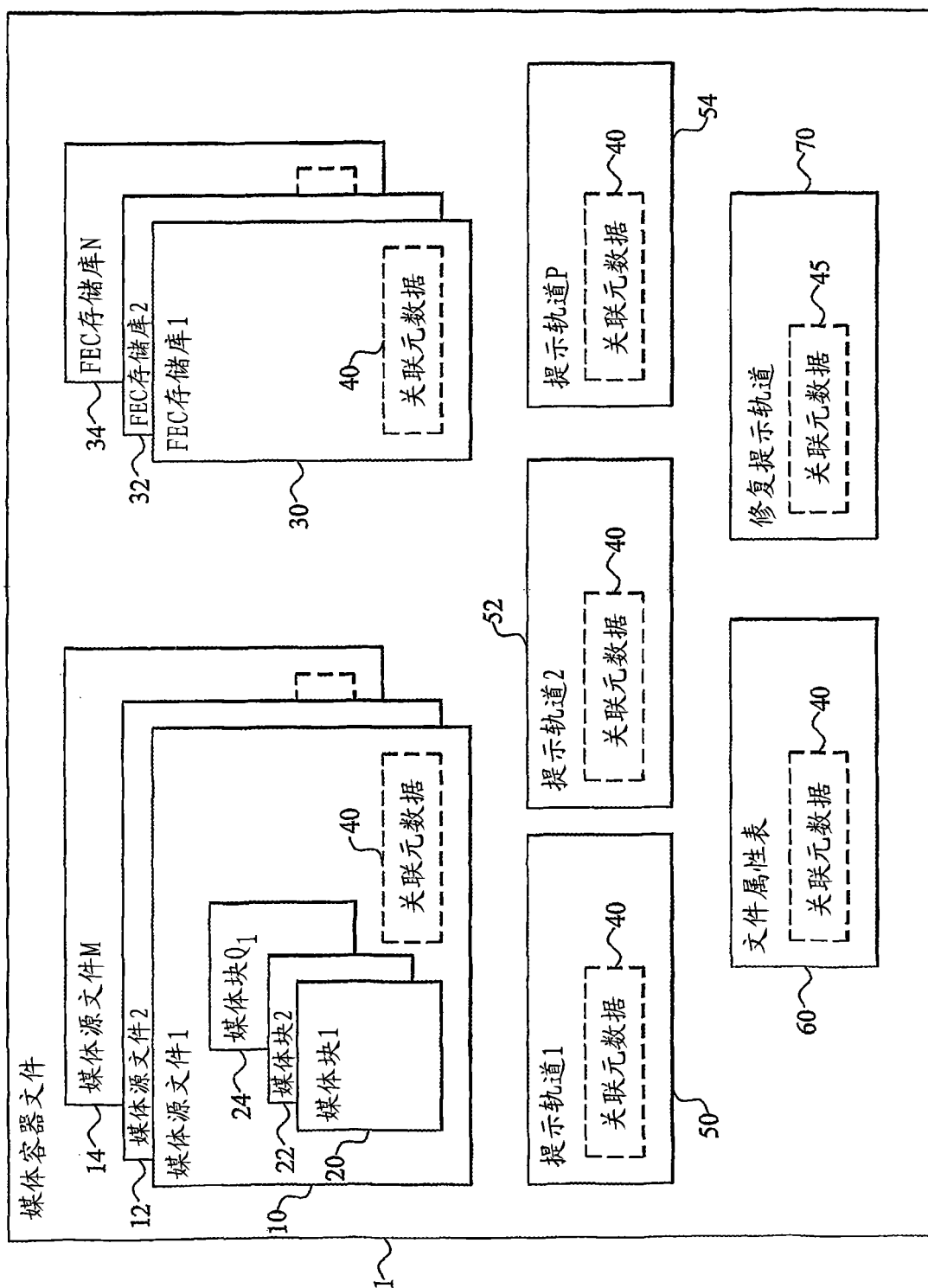


图 11

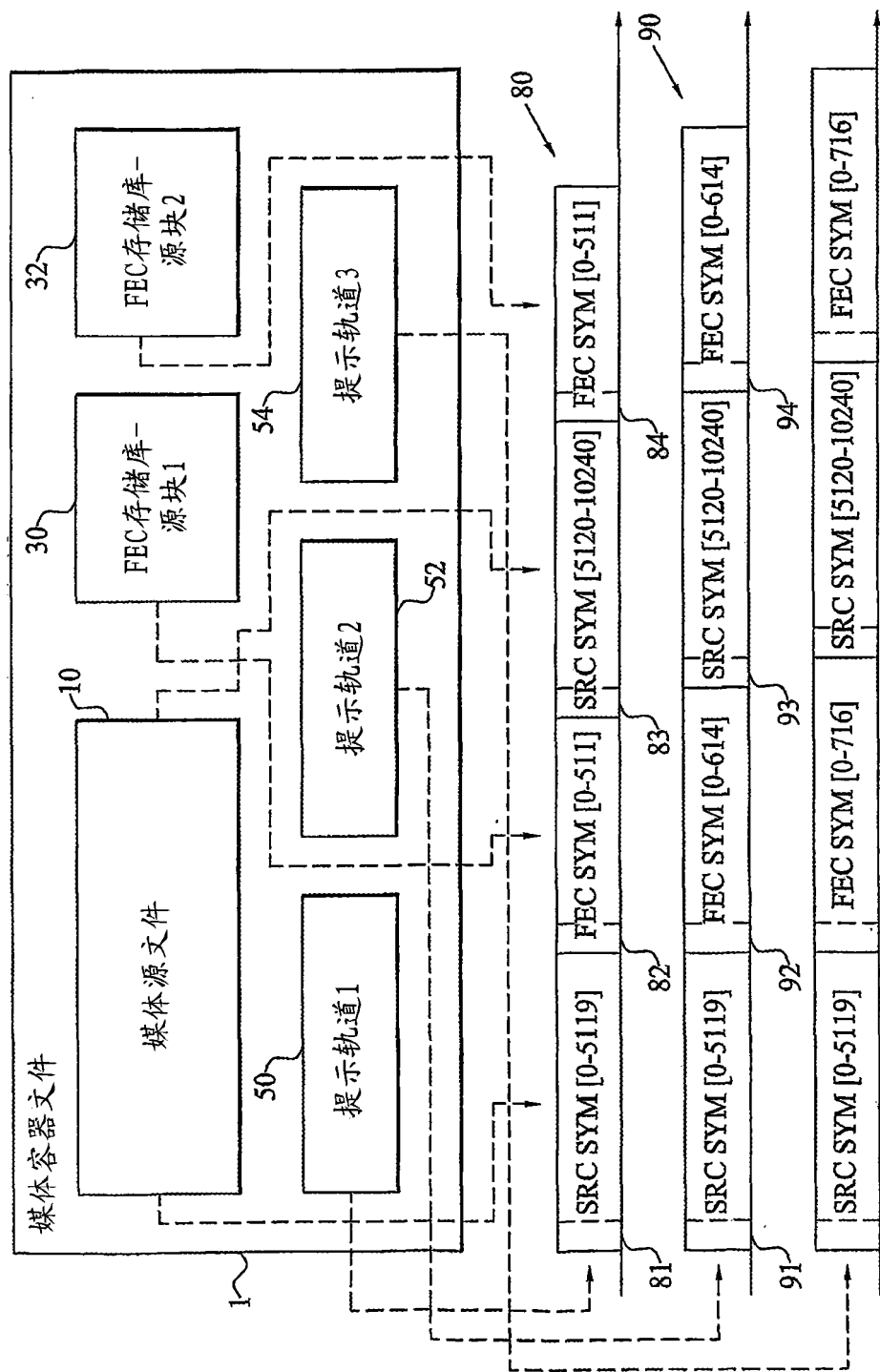


图 12

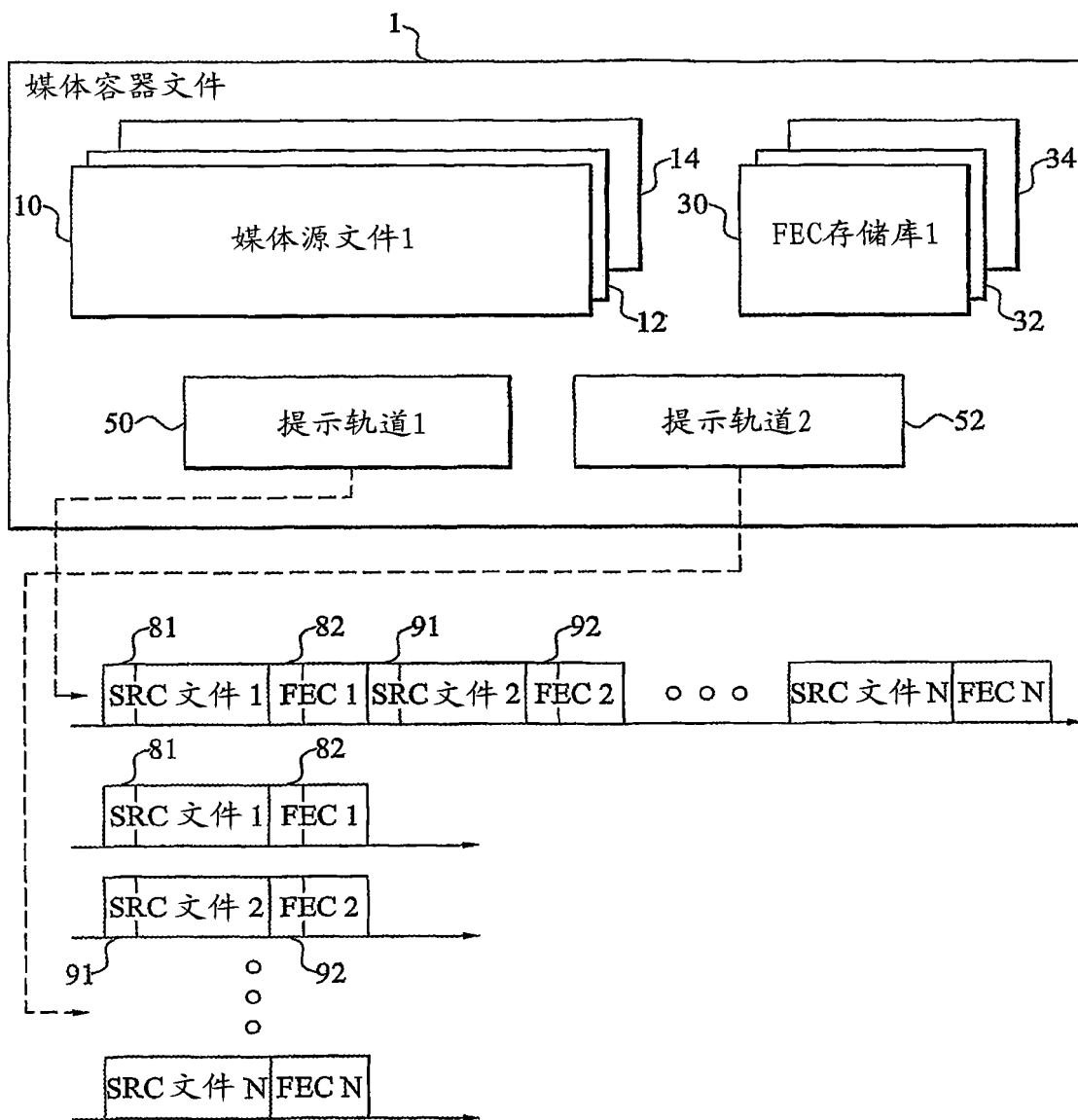


图 13

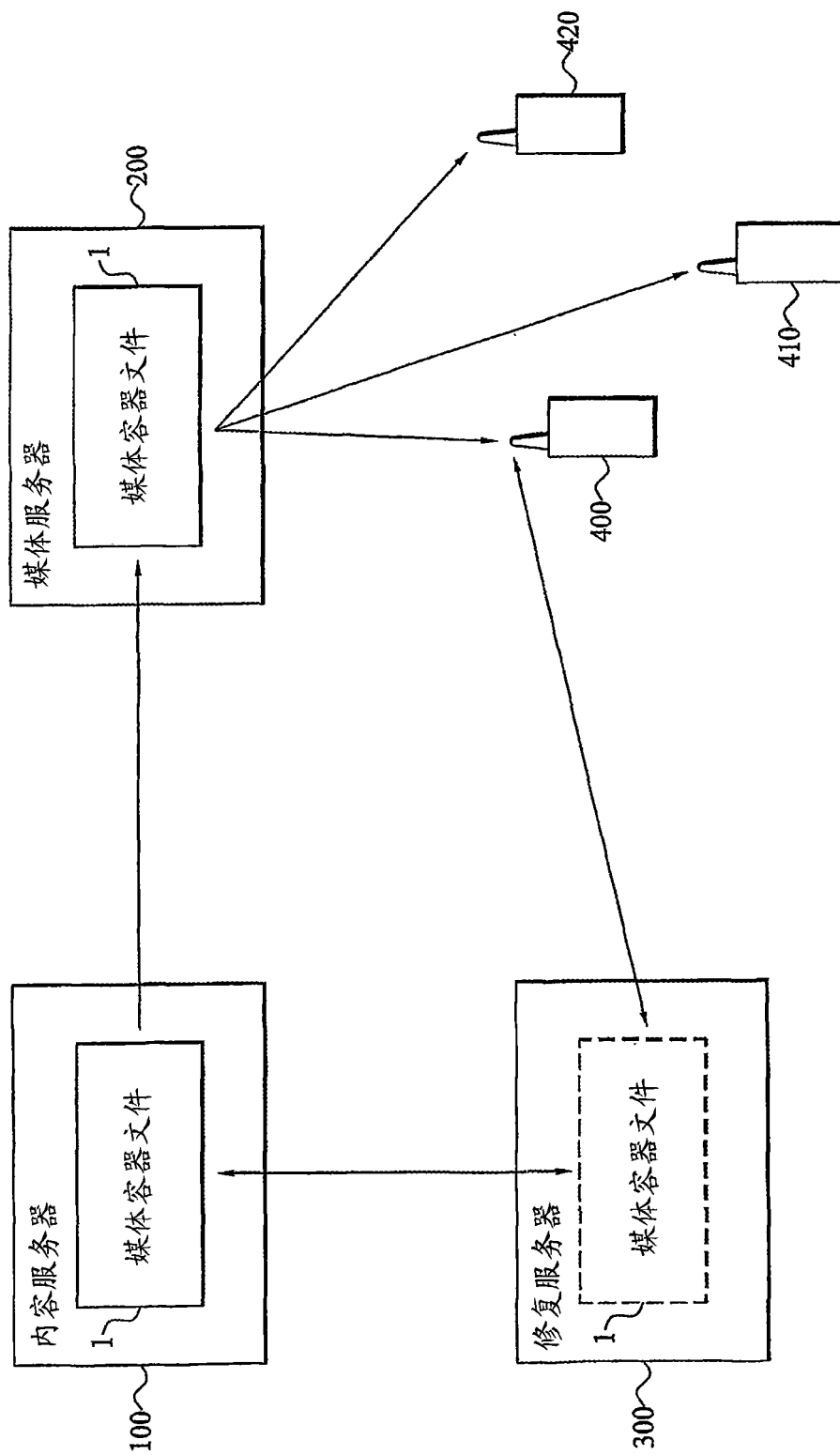


图 14

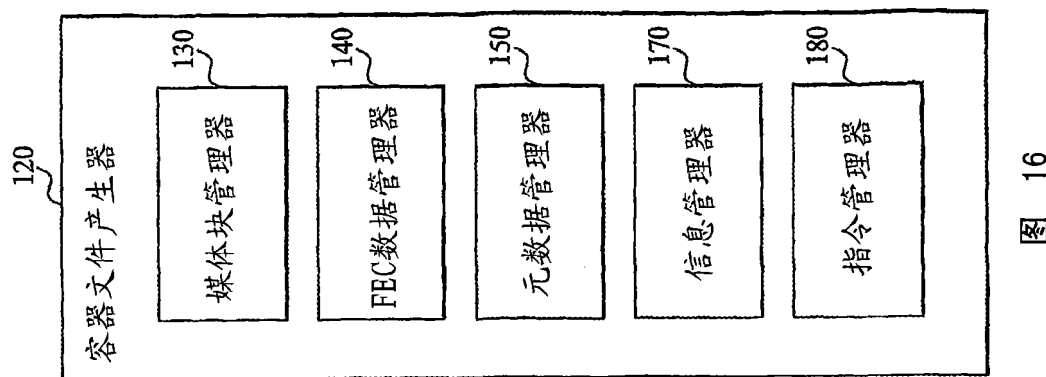


图 16

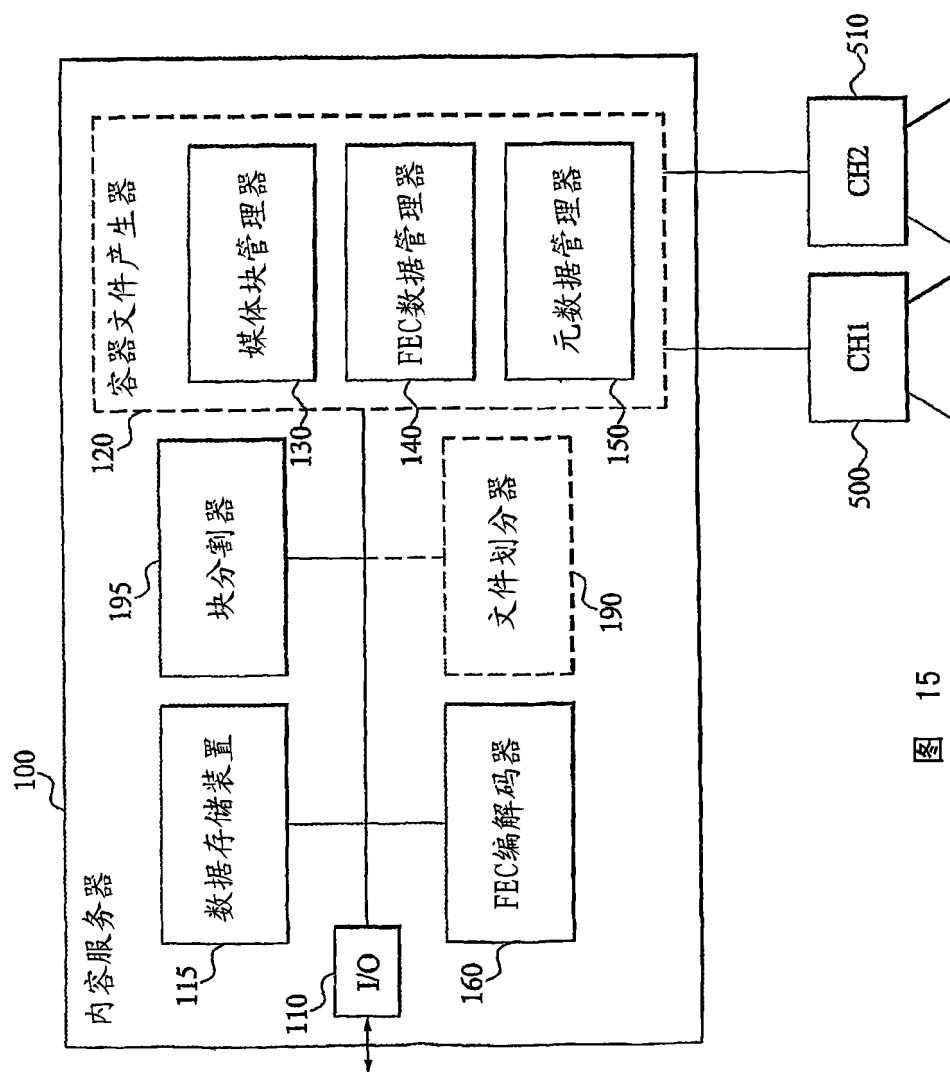


图 15

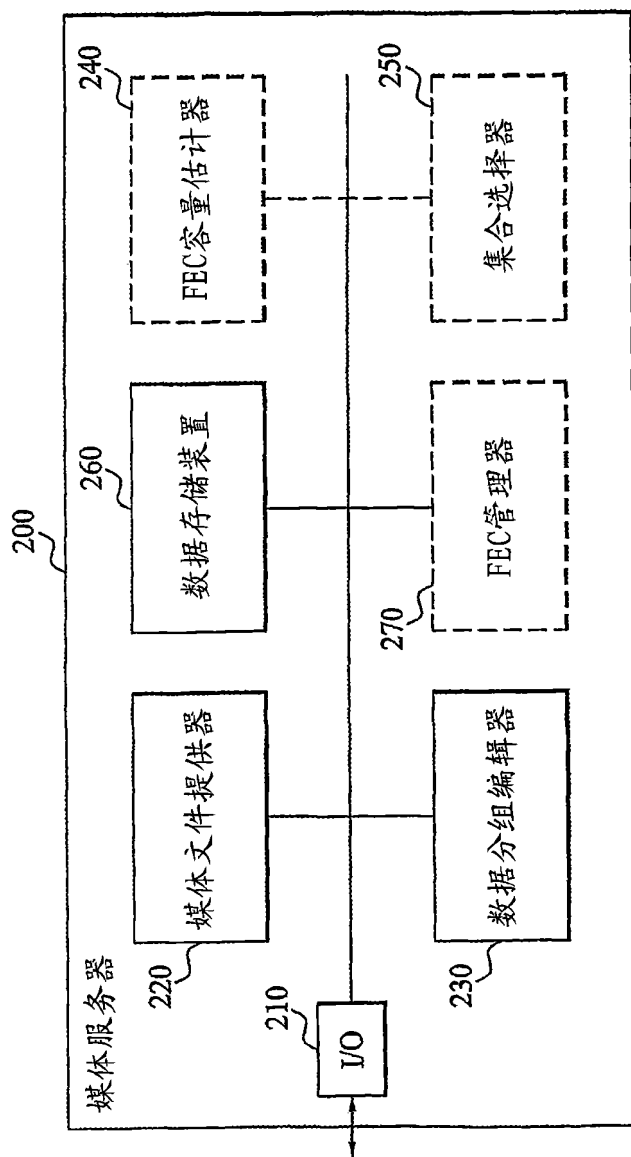


图 17

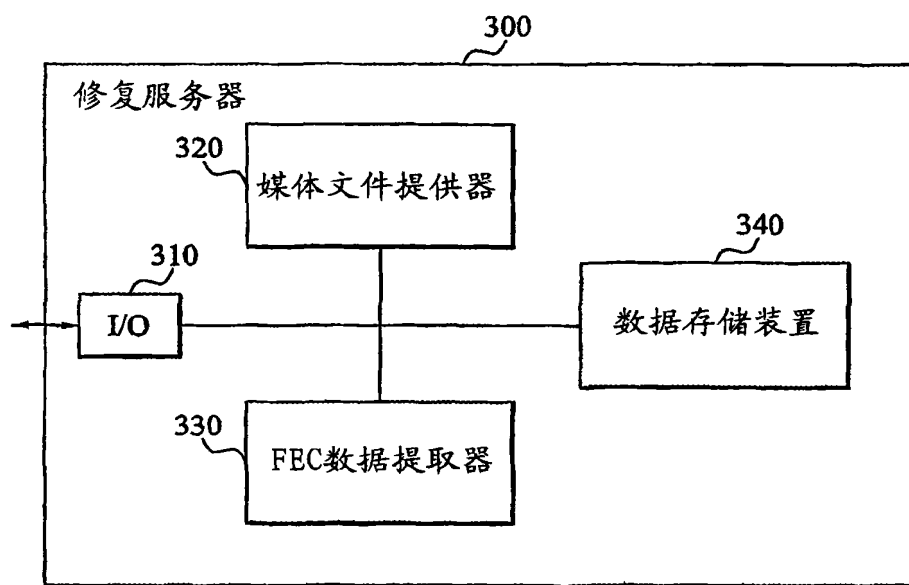


图 18